A Constraint Equation Algebra as a Basis for Haptic Rendering

Matthew Hutchins
CSIRO Mathematical and Information Sciences [6]
GPO Box 664, Canberra, ACT 2601, Australia.
Matthew.Hutchins@cmis.csiro.au

Abstract

Many haptic rendering problems can be expressed in terms of constraints on the motion of a proxy within a virtual environment. This principle is well established for surface rendering, and can also be applied to other types of haptic interaction. A key problem in general constraint based rendering is combining constraints from several sources into a single unified constraint. This paper describes some work in progress toward developing a mathematical framework for manipulating motion constraint equations, and in particular the derivation of a combination algebra for constraints. This work could lead to a system for 6DOF rendering involving non-trivial proxy shapes.

Keywords: haptics, constraints, proxy, 6DOF.

1. Introduction

In a previous PUG paper [4] (and see also [2,3]) we described an approach to haptic rendering based on the use of constraints, and in particular the use of a constrained *proxy*. The use of a proxy (or god-object) for haptic surface rendering is described in [5], and is now a well established technique. The basic idea is that the physical movements of the PHANToM are tracked by a virtual proxy object moving in a virtual environment, as shown in Figure 1. Whereas the PHANToM moves freely in space, the proxy object will encounter virtual objects and fields which will constrain or change its motion. The difference between the free motion of the PHANToM and the constrained motion of the proxy is used as the basis for generating contact forces. This technique is used for surface rendering by preventing the proxy from moving from one side of a surface to the other. It can also be used for other types of haptic interactions, for example: constraining the proxy to a line or plane. A graphical representation of the proxy will usually be rendered in the user interface to provide multimodal feedback. The graphical and haptic properties of the proxy are not necessarily the same — the graphics may be considerably more elaborate than the haptic rendering, for example. In this paper, "proxy" refers to the haptic representation, which may be as simple as a single point.

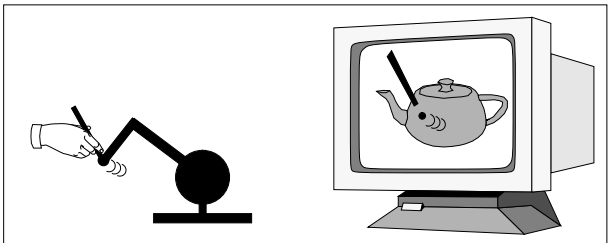


Figure 1: PHANToM and proxy.

The two major haptic rendering SDKs/APIs (Software Development Kits or Application Programming Interfaces) currently available (that we know of) are GHOST from SensAble [8] and Magma from Reachin [7]. Both provide separate facilities for shape based surface rendering and abstract haptic effect or force field rendering. Only the surface rendering interfaces allow manipulation of the proxy position (in GHOST called the Surface Contact Point or SCP). Thus to program a haptic constraint using the proxy technique, one must implement the constraint using the surface interface, or implement a second proxy. The second proxy solution is messy, and doesn't integrate well with surfaces that use the built-in proxy. The surface interfaces are, naturally enough, good for effects that act like surfaces, but awkward to use for more general effects.

The context of the work described in this paper, then, is the development of a new approach to specifying haptic rendering that unifies surface and other types of rendering into a single framework based on constraining the motion of a proxy. This is work in progress, with the majority of the framework still under development. This paper describes some of the mathematical formalisms that have been developed so far for specifying and manipulating constraints on the motion of a proxy.

To provide slightly more context for the mathematics, consider the problem of implementing a haptic scenegraph object, which could be a solid shape, a deformable shape, or some abstract force field. At each traversal of

the scenegraph, the object may be required to solve the problems of collision detection and contact registration. Contact registration means, upon detecting a collision, registering the contact with the rendering system. The contact is the focus of a two-way communication between the scenegraph object and the rendering system. The system must combine the effects of all registered contacts to produce a new proxy position, an output force, and dynamic information to feed back to the scenegraph objects to update their internal state (e.g. deform). One aspect of the contact is the local topology in a neighbourhood of the contact point. This can be expressed as constraints on the motion of the proxy around the contact point. It is essential that the topological information from separate objects, which act independently of each other, can be combined by the system to give a single result. This is the motivation for the algebraic treatment described later in the paper.

2. Motion and constraints

We will assume that the proxy is a rigid body, and its motion is described by rigid body kinematics [1]. At any instant in time, the configuration of a body in space can be described by its position and orientation with respect to some fixed reference "origin". If the configuration of a body in motion is sampled at discrete times, the difference between any two such configurations can be represented[1] by a tuple

$$(A, \hat{\underline{a}}, x, \omega)$$

where A is a point, $\hat{\underline{a}}$ is a unit vector, x is a distance and ω is an angle. Here A is called the anchor point, and together with $\hat{\underline{a}}$ defines a line which is an axis of rotation. The difference in configurations can be interpreted as the effect of a translation of the body along the axis by a distance x, and a rotation around the axis counterclockwise by the angle ω.

We assume that the motion of the PHANToM is an arbitrary continuous motion sampled at discrete times. We wish to approximate this motion over a sampling interval by a simple substitute motion that is easily represented. We choose a form of screw motion where the translation distance and rotation angle change proportionally over time. That is, the difference between two configurations over an interval t will be

$$(A, \hat{\underline{a}}, xt, \omega t)$$

for given constants A, $\hat{\underline{a}}$, x and ω. Thus the complete motion can be represented by a tuple $(A, \hat{\underline{a}}, x, \omega)$ which can be directly computed from the difference between the initial and final configurations of the PHANToM over a sampled interval.

As a body moves in space over time, a point P on the body will move through a curve or trajectory in space which we can describe as $P(t)$. For the proportional screw motion $(A, \hat{\underline{a}}, x, \omega)$, and taking $t \in [0, 1]$, the trajectory of a point is given by the equation

$$\begin{aligned} P(t) \;=\; & P_0 + (-\underline{r}) + xt\hat{\underline{a}} + \underline{r}\cos(\omega t) + \\ & (\hat{\underline{a}} \bullet \underline{r})\hat{\underline{a}}[1 - \cos(\omega t)] + (\hat{\underline{a}} \times \underline{r})\sin(\omega t) \end{aligned} \tag{1}$$

where P_0 is the initial position of P and $\underline{r} = (A - P_0)$. The tangent of the trajectory at any point is given by the derivative of the curve at that point with respect to time:

$$\begin{aligned} P'(t) \;=\; & x\hat{\underline{a}} - \omega\underline{r}\sin(\omega t) + \\ & \omega(\hat{\underline{a}} \bullet \underline{r})\hat{\underline{a}}\sin(\omega t) + \omega(\hat{\underline{a}} \times \underline{r})\cos(\omega t) \end{aligned} \tag{2}$$

This tangent can be thought of as describing the direction that the point P is moving in at time t. Initially, at time $t = 0$, the tangent is

$$P'(0) \;=\; x\hat{\underline{a}} + \omega(\hat{\underline{a}} \times (A - P_0)) \tag{3}$$

We call $x\hat{\underline{a}}$ the translational component of the tangent and $\omega(\hat{\underline{a}} \times (A - P_0))$ the rotational component.

Our goal is to be able to express constraints on the complete motion of the proxy as the combined effect of simple constraints on the motion of individual points in the proxy. One way these constraints would arise is if a point on the surface of the proxy was in contact with the surface of an object in the scene. A constraint on the motion of a point can be expressed as a constraint on the tangent of the trajectory of the point under the motion. For the remainder of this paper we will make two further simplifying assumptions:

- we express motion constraints as constraints on the tangent $P'(0)$ at the start of a sampling interval only, not on the trajectory over the interval;

1. This representation is possible due to a corollary to Euler's theorem that Goldstein [1, p.163] attributes to Chasles.

- we wish to constrain the translational component and the rotational component of the tangent separately.

A more general approach will be left for future work. However, this simplified version may serve as a good enough approximation for the purposes of haptic rendering.

3. Constraint equation algebra

To recap, we have a representation for a particular class of rigid body motions as tuples of the form

$$M = (A, \hat{\underline{a}}, x, \omega)$$

and we have an equation (Eqn. (3)) that describes a tangent of the trajectory of a point under such a motion. By specifying a condition that must be satisfied by the tangent of some point P, we can identify a set of motions that will make the tangent satisfy the condition. Thus a *constraint equation* on the tangent defines a set of legal motions. Clearly, there is a wide variety of conditions that can be placed on the tangent. We wish to choose a useful subset of these conditions and develop a mathematical and computational framework for manipulating them. In other words, we wish to develop a *constraint equation algebra*.

An algebra, in the most general sense as used in algebraic software specification, is simply a collection of sets and functions and relations that satisfy some chosen axioms. To specify an algebra, we need to define what sets there are, what elements are in them, and what operators and relations act on those sets. This is analogous to defining an abstract data type in software. In this case, we wish to define a set of constraint equations, and a single operator to combine pairs of constraint equations. We will want the operator to be idempotent, commutative and associative, so the resulting algebra will have the form of a semi-lattice.

4. The base cases

We start by defining a set of constructors, or generators, or "base cases" for the set of constraint equations. These are the building blocks which will be combined to create the complete set. As we have described, we wish to define these mostly in terms of constraints on the tangent of a particular point at the start of an interval. A useful set of base cases is:

- Free : free motion of the body.
- $\mathsf{PerpTan}(P, \hat{\underline{n}})$: the tangent of P at $t = 0$ is perpendicular to unit vector $\hat{\underline{n}}$.
- $\mathsf{ParaTan}(P, \hat{\underline{n}})$: the tangent of P at $t = 0$ is parallel to unit vector $\hat{\underline{n}}$.
- $\mathsf{FixPoint}(P)$: the tangent of P at $t = 0$ is zero, so that P is fixed.
- Fixed : the entire body is fixed.

We can identify the set of motions permitted by each of these cases by using the tangent equation Eqn. (3). First, some notation for parallel and perpendicular vectors. We define

$$\begin{aligned} \underline{u} \parallel \underline{v} &\equiv (\underline{u} \times \underline{v} = \underline{0}) \\ \underline{u} \perp \underline{v} &\equiv (\underline{u} \bullet \underline{v} = 0) \end{aligned} \tag{4}$$

Then, remembering the assumption that the translation component and the rotation component will be constrained independently, we can derive the following definitions:

$$(A, \hat{\underline{a}}, x, \omega) \in \mathsf{Free} \equiv \mathsf{TRUE} \tag{5}$$

$$\begin{aligned} (A, \hat{\underline{a}}, x, \omega) \in \mathsf{PerpTan}(P, \hat{\underline{n}}) \equiv \\ \{(x = 0)\ \mathsf{OR}\ (\hat{\underline{a}} \perp \hat{\underline{n}})\}\ \mathsf{AND} \\ \{(\omega = 0)\ \mathsf{OR}\ ((\hat{\underline{a}} \times (A - P_0)) \perp \hat{\underline{n}})\} \end{aligned} \tag{6}$$

$$\begin{aligned} (A, \hat{\underline{a}}, x, \omega) \in \mathsf{ParaTan}(P, \hat{\underline{n}}) \equiv \\ \{(x = 0)\ \mathsf{OR}\ (\hat{\underline{a}} \parallel \hat{\underline{n}})\}\ \mathsf{AND} \\ \{(\omega = 0)\ \mathsf{OR}\ ((\hat{\underline{a}} \times (A - P_0)) \parallel \hat{\underline{n}})\} \end{aligned} \tag{7}$$

$$\begin{aligned} (A, \hat{\underline{a}}, x, \omega) \in \mathsf{FixPoint}(P) \equiv \\ \{x = 0\}\ \mathsf{AND} \\ \{(\omega = 0)\ \mathsf{OR}\ (\hat{\underline{a}} \parallel (A - P_0))\} \end{aligned} \tag{8}$$

$$\begin{aligned} (A, \hat{\underline{a}}, x, \omega) \in \mathsf{Fixed} \equiv \\ \{x = 0\}\ \mathsf{AND}\ \{\omega = 0\} \end{aligned} \tag{9}$$

The zero motion that satisfies Fixed will satisfy all of the other constraints. A motion that satisfies $\mathsf{FixPoint}(P)$ must be a pure rotation around an axis through P, and will satisfy any other constraint on P.

5. Composition

We now define a composition operator on constraint equations, denoted $C_1 \oplus C_2$. The motions that satisfy the combination $C_1 \oplus C_2$ should be precisely those that satisfy both of the constraints C_1 and C_2. Thus we define

$$M \in (C_1 \oplus C_2) \equiv (M \in C_1)\ \mathsf{AND}\ (M \in C_2) \tag{10}$$

Another way of saying this is that $C_1 \oplus C_2$ is the intersection of the sets C_1 and C_2. Thus, we know this com-

position operator satisfies the axioms required of a semi-lattice (idempotency, commutativity, associativity), because set intersections do.

The complete set of elements in the algebra is therefore all those generated by the five base cases, plus the composition of any two other elements. This is essentially a recursive definition, and computationally would require a recursive data structure to represent the elements. However, we can make some observations to simplify this. Firstly, suppose that all constraints must be applied to the same point P. This would be the case for a 3DOF rendering system with a single point proxy, or, such as in Magma, a small spherical proxy where all constraints are translated to apply to the centre of the sphere. It turns out that the five base cases completely characterise the system. That is, every combination of two or more constraints applied to the same point are equivalent to a simple constraint applied to the same point. Mostly, the result of $C_1 \oplus C_2$ is either C_1 or C_2 or $\mathsf{FixPoint}(P)$. The only interesting case is

$$\mathsf{NOT}(\hat{\underline{n}}_1 \parallel \hat{\underline{n}}_2) \Rightarrow$$
$$(\mathsf{PerpTan}(P, \hat{\underline{n}}_1) \oplus \mathsf{PerpTan}(P, \hat{\underline{n}}_2) = \mathsf{ParaTan}(P, \hat{\underline{n}}_1 \times \hat{\underline{n}}_2))$$

For the general case where constraints can be applied to different points on the proxy, there are definitely more elements required. However, it appears that the five base cases plus the six pairwise combinations of the non-trivial base cases will be enough to completely characterise the space. So, all constraints could be represented in a flat data structure with eleven types of elements. The proof of this conjecture is work currently in progress.

6. Future work

The combination algebra developed above makes it possible to reduce a set of independent constraints to a single constraint that must be satisfied by the motion of the proxy. Given a potential motion and a constraint, there is a straightforward decision procedure to determine if the motion satisfies the constraint. However, if a motion of the proxy does not satisfy the constraint, it is necessary to find an alternative motion that does. This is always possible (the zero motion satisfies any constraint), so in fact the problem is to find the "best" alternative motion. The optimal solution may be different for each of the eleven constraint cases (of course, the Fixed and Free cases are easy!) Once the development of the algebra is complete, this will be the next problem to be solved. Some cases have been solved already. We assume that, at present, most haptic rendering will be done using a 3DOF output device, or a 6DOF device where the rotational fidelity is less than the translational fidelity. Therefore it is best to try to match the rotation component of the motion as closely as possible. This will minimise the discrepancy between the orientation of the haptic device and the orientation of the proxy, which will minimise the required torques. Note that the screw motion representation naturally isolates the translation and rotation components of the input motion.

As described in the introduction, this formulation of constraint equation algebra is just one part of a larger specification for a new approach to haptic rendering. Our future work will be the continued development and implementation of this approach.

References

[1] Herbert Goldstein. Classical Mechanics. Second edition, Addison-Wesley, 1980. Chapter 4, pages 128–187.

[2] Chris Gunn and Paul Marando. Haptic constraints: guiding the user. In proceedings of SimTecT '99, Melbourne, Australia, March 1999. Pages 261–264.

[3] Matthew Hutchins. Software components for haptic constraints. In proceedings of the SPIE Vol. 3957, Stereoscopic Displays and Virtual Reality Systems VII, 2000. Pages 423–432.

[4] Matthew Hutchins and Chris Gunn. A haptic constraints class library. In Salisbury, J.K. and Srinivasan, M.A. (Eds), Proceedings of the Fourth PHANTOM Users Group Workshop, AI Lab Technical Report No. 1675 and RLE Technical Report No. 633, MIT, November 1999.

[5] C.B. Zilles and J.K. Salisbury. A constraint-based god-object method for haptic display. In proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems, August 1995. Pages 146–151.

Web Cites

[6] CSIRO Mathematical and Information Sciences. http://www.cmis.csiro.au

[7] ReachIn Technologies. http://www.reachin.se

[8] SensAble Technologies. http://www.sensable.com

Fuzzy Rule-Based Evaluation for a Haptic and Stereo Simulator for Bone Marrow Harvest for Transplant

Liliane dos Santos Machado [(1)]
Ronei Marcos de Moraes [(1,2)]
Marcelo Knorich Zuffo [(1)]

[(1)] Laboratório de Sistemas Integráveis - Universidade de São Paulo
São Paulo – SP – Brazil
{liliane, ronei, mkzuffo}@lsi.usp.br

[(2)] Statistics Departament - Federal University of Paraiba
João Pessoa – PB – Brazil
ronei@de.ufpb.br

Abstract. Virtual Reality has been used to simulate procedures in several fields, especially those where critical tasks are involved as simulation of some invasive medical procedures. We are developing a low-cost haptic and stereo simulator for bone marrow harvest for transplant. The system includes an intelligent evaluation procedure that allows classify the trainee learning. The present paper describes the proposed system, details of its implementation and results we just obtained.

Introduction

Bone marrow transplant, despite commonly held perceptions, is not a usual surgery. Basically, the bone marrow transplant consists of an infusion of healthy cells, capable of generating identical copies of themselves and producing blood cells. This blind invasive procedure is relatively simple, but the success of the procedure will depend on the physician's dexterity, and his ability to manipulate the needle in a complex anatomical region.

This work presents a virtual reality system to simulate bone marrow harvest for transplant. Bone marrow transplant, despite commonly held perceptions, is a semi-invasive procedure that depends on the physician's dexterity, once there is no visual information of the patient body internal structure.

In this simulator we are using a haptic device and stereo view glasses to give an immersion degree satisfactory to the user trainee. To give an evaluation of the training we add to the system an intelligent evaluation tool based on fuzzy rules. This way, the expert doctor knowledge is modeled by fuzzy rules by four variables to give a classification of the procedure performed by the trainee.

Motivation

The bone marrow transplant is a relatively new medical procedure to treat recently considered incurable diseases. The first success transplant was made in 1968, and since then has been a current procedure for patients with leukemia, aplastic anemia, lymphomas, multiple myelomas, disturbs in the immunology system and in some solid tumors such as the breast cancer and ovarian cancer [Oncolink, 1999].

The process to extract the bone marrow is made through many material aspirations from the iliac crest bone marrow (sometimes it includes the sternum bone also) from the donator under general anesthesia. The procedure is a blind procedure without any visual feedback except the external view of the donor body, the physician need to feel the skin and bone layers trespassed by the needle to find the bone marrow and then start the material aspiration. From the physicians point of view the bone marrow harvest demands great ability, which will offer a better recovery to the donator and less pos-harvesting pain. Particularly on children the bone marrow harvest for transplant is critical considering that bones in this case are thin and soft, and that the patient receive a smaller degree of anesthesia. The Children's Institute of Hospital das Clínicas de São Paulo - Brazil realize on average 15 procedures every year. Currently the only training procedure available for novice doctors is training with guinea pigs, real procedure observation and further supervision by physicians in real procedures.

In the same way, it is known that expert physicians evaluate trainee learning observing the needle position, its angle when inserted in the body of the patient and how deep it is, beyond the fact the trainee extract the bone marrow (goal of the procedure). That means, the expert doctor evaluation is partially subjective. One way to model subjective knowledge is using fuzzy sets [Dubois and Prade, 1980]. In our system physicians knowledge in bone marrow harvesting is modeled by fuzzy rules by the four variables described bellow (needle position, angle, depth and bone marrow extraction) to give a classification of the procedure done by the trainee.

With our system, we intend to improve the learning for novice doctors once the dexterity necessary will be acquired through the training on a virtual reality system, which will simulate and evaluate the procedure with a force-feedback device integrated. We expect to reduce and improve the learning curve affecting donors' rehabilitation.

Intelligent Evaluation Systems

Intelligent evaluation is called the one made by an expert. In intelligent evaluation systems, the expert knowledge is stored in a knowledge database using some logic representation, generally by rules. The rules utilization control is done by an inference system. The architecture formed by the knowledge database and the inference system is called "expert system". When there is subjectivity in the knowledge database its representation can be done by fuzzy models [Zadeh, 1988], where the subjectivity is modeled by fuzzy sets.

In fuzzy expert systems the expert's knowledge about a specific task is modeled by fuzzy rules. The variables of interest are collected by a subsystem and sent to the expert system. The data provided by the variable values will be analyzed by the rules database. Each rule is related to a variable of interest and each expert can have his own conclusion about a specific fact. The several conclusions about one rule are aggregated to compose a new fact. This fact will be analyzed by the rules set providing a conclusion about the facts presented.

For the evaluation, we utilize an expert system joined to the simulator. To evaluate a trainee, the expert knowledge about the procedure is modeled by fuzzy rules. The variable values are "collected" by the haptic device and sent to the expert system to be analyzed by the rule database. The pertinent facts of a rule are joined and analyzed to provide a final classification of the trainee. We are using five types of fuzzy

classifications to a trainee: *you need much more training*, *you need more training*, *you need training*, *your training is good* or *your training is excellent*, allowing identify if more training is or is not necessary.

Proposed System and Implementation

The proposed system is a semi-immersive virtual reality system [Pimentel, 1995] where the trainee and some expectators doctors (tutor and trainee) can share the same stereoscopic view of the bone marrow harvest procedure simulation [Machado, 2000]. A high end PC Pentium III 600Mhz platform with and AGP 3Dlabs Oxygen GVX1 board including a time-multiplexed Stereo Graphics Crystal Eyes shutter glasses [Stereographics, 1997] and a Phantom Desktop haptic device composes our simulator [Sensable, 1999 and Sensable, 2000]. The simulator consists in a force feedback virtual interactive model of tissue layers from the pelvis region and its hardness and texture characteristics.

Using a virtual syringe with tactile feedback (simulated by the Phantom Desktop) the user can penetrate thought the several tissue layers feeling the transitions among tissues, as well as feeling the texture associated to each layer.

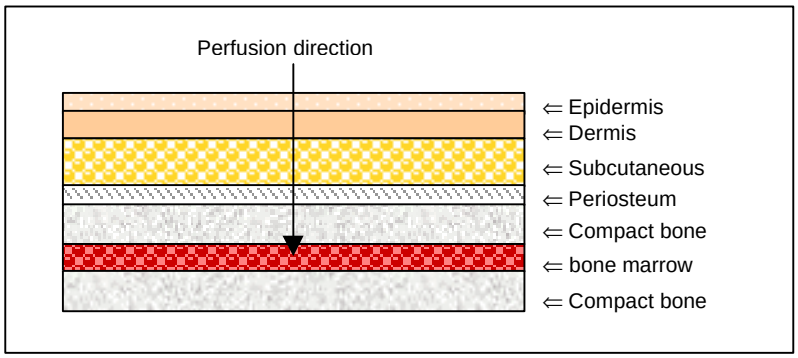


Figure 05 - The Perfusion Tissue Layers

The feedback sensation associated to each layer was modeled from tactile sensation descriptions done by specialists. So, we modeled the several physical properties of the tissues in the iliac crest in the following layers:

- Epidermis: approximately 2 mm thick, elastic and slippery tissue;
- Dermis: approximately 7 mm thick, elastic tissue;
- Subcutaneous: approximately 4 mm thick, soft and non-resistant tissue;
- Periosteum: approximately 2 mm thick, resistant, slippery, lubricated and smooth tissue.
- Compact bone: approximately 5 mm thick, hard and resistant tissue;
- Bone marrow: approximately 10 mm thick, soft tissue, without resistance.

The fuzzy rules of our expert system are modeled by membership functions according to specifications of experts. Several types of membership functions can be used as

trapezoidal, triangular and pi-functions and the fuzzy inference system used is Mamdani-type [Mamdani, 1975]. An example of rule for this expert system is:

IF Position_x is *left_center* AND Position_y is *up_center* AND Position_needle is *acceptable* AND Marrow_harvest is *yes* THEN Trainee_class is *you_need_training*

where: Position_x, Position_y are coordinates which the needle touch the patient body; Position_needle is the angle of needle input to body of patient; Marrow_harvest shows the success or failure of trainee to harvest bone marrow and Trainee_class is the classification of trainee.

Acknowledgements

This project is funded by Fundação de Amparo à Pesquisa do Estado de São Paulo, grant # 99/01583-0, with additional support from FINEP – Brazilian Federal Fundation Agency.

Thanks to Dr. Andre Nebel de Mello and Dr. Vicente Odone Filho of Children's Institute of Hospital das Clínicas for their helpful insights about bone marrow harvesting.

Bibliography

Dubois, D.; Prade, H. Fuzzy Sets and Systems: Theory and Applications. Academic Press, 1980.

Machado, L. et al.; A Virtual Reality Simulator for Bone Marrow Harvest for Transplant. Proceedings of 3th Brazilian Workshop of Virtual Reality, October, 2000.

Mamdani, E. H.; Assilian, S.; An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller, International Journal of Man-Machine Studies, vol.7, pp. 1-13, 1975.

Oncolink; URL from the Cancer Center at the University of Pennsylvania (USA). http://cancer.med.upenn.edu/specialty/med_onc/bmt/, August, 1999

Pimentel, K.; Teixeira, K. Virtual Reality - through the new looking glass. 2.ed., McGraw-Hill, 1995.

Sensable; Ghost SDK Programmer's Guide Version 3.0. Sensable Technologies Inc., March 1999.

Sensable; Ghost API Reference Version 3.0. Sensable Technologies, Inc. March 2000.

Stereographics; Developer Handbook, StereoGraphics Corp., 1997.

Zadeh, L.A.; Fuzzy Logic, Computer, vol.1, pp. 83-93, 1988.

Haptic Collaboration over the Internet

João P. Hespanha, Margaret McLaughlin, Gaurav S. Sukhatme
Minoo Akbarian, Rajiv Garg, Weirong Zhu

Integrated Media Systems Center
University of Southern California
Los Angeles, CA 90089

Abstract

We address the real-time collection and simultaneous broadcast of haptic information to multiple haptic session participants, so that collaborative exploration of objects is possible, even when users are equipped with disparate haptic devices, such as the PHANToM and the CyberGrasp. We have designed and are currently testing a prototype system for haptic collaboration over the Internet. The basic idea is to provide a framework for multiple users (each with his or her own haptic device connected to a computer) to share a common experience of touch. This will allow users to exert forces on each other through the network as well as exert forces on common objects.

In this paper we present a distributed architecture for haptic collaboration via the Internet. We explicitly address the issue of latency (communication delay), thus providing a foundation for a shared haptic experience among distributed users. With respect to stability, latency is a critical factor that governs whether two users can truly share a common haptic experience. We propose an algorithm where the nature of the interaction between two hosts is decided dynamically based on the measured network latency between them. Users on hosts that are near each other (low communication latency) are dynamically added to fast local groups. If the communication latency is high, users are allowed a slower form of interaction where they can touch and feel objects but cannot exert forces on them. Users within a fast local group experience true haptic collaboration since the system is able to resolve the interaction forces between them fast enough to meet stability criteria. We discuss the creation, maintenance and update mechanisms of local groups for fast interaction, as well as synchronization mechanisms for hosts participating in slower interaction. We conclude with a discussion of open issues and future work.

1. Introduction

Haptic (adj): of or relating to the sense of touch. In the present context, haptic refers to the modality of touch and the sensation of shape and texture an observer feels when exploring an object in a virtual environment. Applications of haptics include online museums [6], aid for the visually impaired, remote surgery and entertainment. In many of these applications it will be necessary for users to **interact with each other** as well as with other objects. In this article, we propose an architecture for haptic collaboration among distributed users. We focus on collaboration over a non-dedicated channel (such as an Internet connection) where users experience stochastic, unbounded communication delays [7].

The area of haptic collaboration is relatively new. There have been a few prior studies that we briefly review here. In a study by Basdogan et. al. [1], partners at remote locations were assigned three cooperative tasks. Experiments were conducted with visual feedback only, and with both visual and haptic feedback. Both performance and feelings of togetherness were enhanced in the dual modality condition. Durlach and Slater [3] note that factors that contribute to a sense of co-presence include being able to observe the effect on the environment of actions by one's interlocutors, and being able to work collaboratively with co-present others to alter the environment. Buttolo et. al. [4] note that when the same virtual environment is shared between two distributed sites there may be registration problems. Representations of the virtual object must coincide, but the distributed nature of the communication, especially over the Internet, may introduce considerable latency whose effects may be hard to predict.

2. Virtual Haptic World

Imagine you decide to go to a handicraft museum. There is a map of the museum at the door showing different halls in the museum, each containing a group of handicrafts. Upon entry into a hall, you can see the handicrafts and the other people in that room. You can touch all of the objects in the room and interact with them. In a real museum, all of the above are familiar experiences, except for the last one. As a matter of practice, touching art objects is usually strictly prohibited.

The scenario described above motivates the research presented here. Our goal is to design an architecture that will support collaborative touch in virtual environments. We term such environment a ***virtual haptic world***. As shown in *Figure 1*, users may have different kinds of haptic devices, such as the PHANToM, CyberGrasp, or a FEELit mouse, or they can just be viewers. Some of the participants in the haptic world may only provide virtual objects as a service to the remaining users. This would be the role, e.g., of a museum's server.

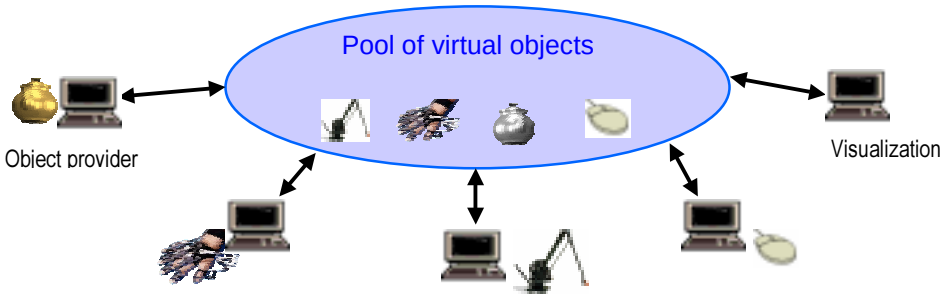


Figure 1: A virtual haptic world

From a computational perspective, a haptic world consists of a network of nodes. Each node corresponds to a computer whose operator is part of the shared virtual environment. The operator will typically interact with virtual objects through a haptic device, but conceivably, some users may interact with the haptic world using other modalities, e.g. by simple visualization. Some nodes may operate autonomously (i.e., without a human operator) and simply provide virtual objects for the haptic world.

Each node in the haptic world contributes to the shared environment with virtual objects. These can be static, e.g., a sculpture "bolted" to the ground, or dynamic, e.g., a teapot that can be virtually manipulated. We view the haptic devices that the human operators use to interact with the haptic world as dynamic objects. Each object in the haptic world is *owned* by one of the nodes, which is responsible for defining how its dynamic properties evolve. Typically, a node that is physically connected to a haptic device owns the object that represents the device.

Two databases are used to represent a haptic world. The *node database* contains information about the node network. It stores the logical identifiers and the IP addresses of all nodes, as well as the latency and available bandwidth between all nodes. The need for this information will become clear later. This database is dynamic because new nodes may join or leave the haptic world at run-time. The *object database* contains the information about all objects that are part of the haptic world. Each record in this database refers to a particular object and it contains the object identifier, the identifier of the node that owns it, its static properties (shape, size, color, etc.) and its dynamic properties (position, orientation, velocity, etc.).

The force control algorithms used for haptic rendering generally require high sampling rates (typically, on the order of 1KHz) and low latency (typically, on the order of a few milliseconds) [5]. This means that the databases need to be queried very frequently and with very low delay. Because of this it is necessary to distribute these databases by keeping local copies at each node. This allows for very fast access to the data about the objects that is needed for the force feedback loops, at the expense of the added complexity introduced by issues related to the consistency between the databases. Much of what follows is precisely related to the problem of keeping the databases synchronized so that all nodes have roughly the same perspective on the shared environment.

3. Database Synchronization

Since the object database contains data that is dynamic, the local copies of this database that exist at each node must be kept synchronized by a periodic exchange of data. This is done by a very simple mechanism that uses the concept of object ownership introduced earlier: periodically, the owner of each object broadcasts the dynamic properties of its objects to all other nodes. Each node must then continuously listen to the other nodes for updates on the dynamic properties of the objects that it does not own. This is represented schematically in *Figure 2*.

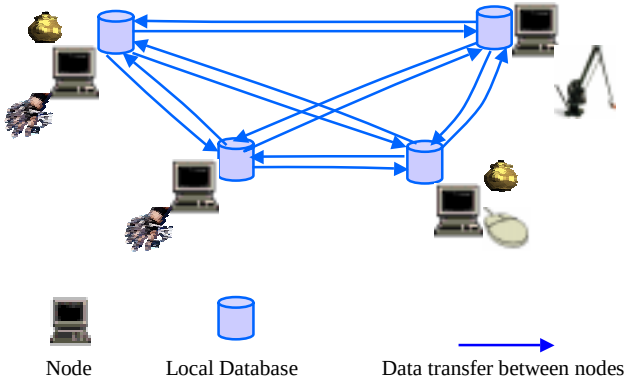


Figure 2: Object database synchronization in the haptic world.

```
While (1)
{
        Listen for dynamic properties of objects owned
            by other nodes and update local object
            database
        Query haptic rendering system for dynamic
            properties of objects owned
        Broadcast dynamic properties of objects owned
        Wait for one synchronization period
}
```

Table 1: Pseudo-code for object database synchronization

Typically, the haptic rendering system uses the following fairly standard algorithm:

```
Compute amount of overlapping between objects owned and all other objects
Compute forces on objects owned (assuming spring-damper system)
Transmit forces to haptic device
Integrate forward in time to predict dynamic properties of objects owned at next sampling time
```

Table 2: Pseudo-code for the update of dynamic properties of objects

When the number of nodes is large, the broadcast of object properties required by the algorithm in *Table 1* may be costly unless the synchronization period is large. We will address this issue later.

Another main challenge arising from the distributed nature of the databases that store the information about the haptic world is related to the addition and removal of nodes from the haptic world. When a new node joins the haptic world, it must first receive the current node and object databases from some other node in the haptic world. It must then add itself to the node database and add its objects to the object database. Finally, it must inform all other nodes of these changes to the databases. This is implemented by the pseudo-code shown in *Table 3* that must run in every node.

```
Request copy of node database
Request copy of object database
Add self to node database
Add objects owned to object database
Broadcast request to add new record to node database
Broadcast request to add new records to object database
While (node active)
{
Listen for requests to:
      send node/object database
      add/remove record to/from node database
      add/remove record for/from object database
}
Broadcast request to remove self from node database
Broadcast request to remove owned objects from object
      database
```

Table 3: Pseudo-code for the creation of a new node in the haptic world

```
Input: G = {list of objects in local group}

While (1)
{
      Listen for the dynamic properties of objects in G owned
          by other nodes and update local object database
      Query haptic rendering system for dynamic properties
          of objects owned
      Broadcast to the owners of the objects in G the dynamic
          properties of objects in G owned by self
      Wait for one local group synchronization period.
}
```

Table 4: Pseudo-code for local group synchronization

4. Local Groups

The broadcast required by the synchronization algorithm in *Table 1* can be very costly when the number of nodes is large. Because of this, the synchronization period may need to be fairly long. For static objects this poses no problems, but the same is not true for dynamic objects, i.e., objects that can move.

When two or more dynamic objects touch each other, the resulting motion must be computed by simulating Newton's laws using an algorithm similar to the one in *Table 2*. However, when the same node does not own all the objects involved in a close interaction, each object only observes the effect of its motion in the motion of other objects at a relatively low sampling rate, determined by the synchronization period. This leads to very unrealistic motions (and possibly instability) because the algorithm in *Table 2* no longer provides a good approximation to Newton's law. We overcome this by creating small groups of nodes that engage in very fast and very frequent exchange of synchronization data for objects in close interaction. The creation of these groups is, of course, only possible when the bandwidth between the nodes is sufficiently large and the latency is sufficiently small. Because of the high cost of local groups, these should only be maintained while the objects are interacting.

As explained above, to resolve the motion of objects involved in close interaction a high bandwidth/low latency synchronization mechanism is needed. In our architecture this is achieved by introducing the concept of a *local group*. A local group consists of a group LG of objects, whose owners enhance the basic synchronization algorithm for those objects in LG, by decreasing the synchronization-sampling period. The local group synchronization algorithm, given in *Table 4*, is very similar to the basic one in *Table 1*.

Since each local group determines the positions of all the objects in that local group, each object should belong to, at most, one local group (this does not prevent a node that owns several objects from being involved in several local groups). Moreover, the fast synchronization within the local group requires high bandwidth and low latency between the nodes involved. Special care must therefore be paid to the creation of local groups.

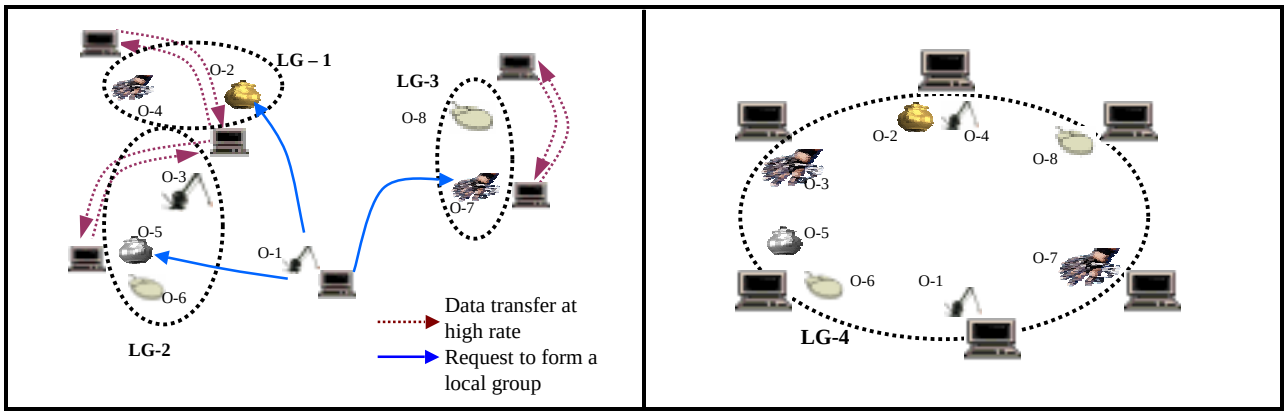


Figure 3: Haptic world with three local groups and a node requesting to create a local group

Figure 4: New local group, after the request in Figure 3 was processed

We use an example to illustrate the issues involved in the management of local groups. Consider the haptic world shown in *Figure 3*. In this figure we see three local groups: LG-1 is formed by the set of objects {O-2, O-4}, LG-2 is formed by {O-3, O-5, O-6}, and LG-3 is formed by {O-7, O-8}. Note that the same node owns the objects O-2 and O-3 but they are part of distinct local groups. This means that, although belonging to the same node, these objects are not in close proximity and therefore their motions are independent. Suppose now that the user at the node that owns O-1 wants to use O-1 to manipulate the objects O-2, O-5, and O-7 (*Figure 3*). This requires the creation of a local group that contains T = {O-1, O-2, O-5, O-7}. However, since some of these objects are already part of other local groups, the old local groups LG-1, LG-2, LG-3 must be destroyed and a new local group LG-4 must be created, containing the objects in T as well as those in the old local groups LG-1, LG-2, and LG-3 (*Figure 4*). This only occurs if the network connections between all the nodes that own the objects in question have sufficiently large bandwidth and sufficiently low latencies for the local group synchronization.

The pseudo-code in *Table 5* implements the algorithm used to create a new local group. The pseudo-code in *Table 3* also needs to be modified as shown in *Table 6* to process the requests generated by the algorithm in *Table 5*.

```
Input: T = {desired list of objects in new local group}

L = Expand (T)      % determine list of all objects that need
                    %  to be included in new local group
If Feasible (L)
{    % Only create local group if all nodes involved
     % satisfy the bandwidth and latency requirements
     For each l ∈ L
          Request owner of object to destroy the local group
               to which it belongs
     For each l ∈ L
          Request owner of object to create a local group for
               objects in L
     Return Success
}
Else
     Return Failure
```

Table 5: Pseudo-code to create a new local group

```
{…}

While (node active)
{
   Listen for requests to:
        send node database
        send object database
        add/remove record to node database
        add/remove record for object database
        create/destroy a local group
}

{…}
```

Table 6: Modification in the pseudo-code in Table 3 to process the requests generated by Table 5.

5. Conclusions and Future Work

We proposed an architecture for the real-time collection and simultaneous broadcast of haptic information to multiple haptic session participants, so that collaborative exploration of objects is possible, even when users are distributed across a network. The architecture relies on two distributed databases: the node and the object databases. These two databases are dynamic and need to be kept coherent among all nodes in the virtual haptic world. We presented pseudo-code for the algorithms that keep these databases synchronized. These algorithms are independent of the actual haptic devices employed by each user.

In future work, we hope to make significant progress on the registration of the haptic display systems in collaborative-networked environments. We will also examine the necessary entities to achieve networked collaboration with disparate haptic devices (pen-based versus glove-based, small versus large workspace). We plan to address not only integration issues but also questions related to the interaction process itself, including feelings of co-presence and performance satisfaction, and how these variables are affected by the exploration modality (vision, vision plus haptic or haptic only). Another line of research is the development of force control algorithms tailored to a distributed haptic environment. These algorithms must be robust with respect to the stochastic delays caused by the communication network.

6. References

[1] C. Basdogan, C. Ho, M. Slater, and M. A. Srinivasan, "The Role of Haptic Communication in Shared Virtual Environments," PHANToM Users Group, 1998 (http://www.sensable.com/community/PUG98/19_basdogan.pdf).

[2] C. Ho , C. Basdogan, M. Slater, N. Durlach, M. A. Srinivasan , "An Experiment on the Influence of Haptic Communication on the Sense of Being Together," 1998 (http://www.cs.ucl.ac.uk/staff/m.slater/BTWorkshop/touchexp.html).

[3] N. Durlach and M. Slater, "Presence in Shared Virtual Environments and Virtual Togetherness," 1998 (http://www.cs.ucl.ac.uk/staff/m.slater/BTWorkshop/durlach.html).

[4] Buttolo, P., J. Hewitt, R. Oboe, and B. Hannaford, "Force Feedback in Virtual and Shared Environments," 1997 (http://brl.ee.washington.edu/BRL/publications/Rep100.ps).

[5] Wilson, J. P, Kline-Schoder, R. J., Kenton, M. A., and Hogan, N, "Algorithms for Network-Based Force Feedback," in Salisbury, J. K. and Srinivasan, M. A. (Eds), Proc. PHANTOM Users Group Workshop, MIT 1999.

[6] McLaughlin, M. L., Sukhatme, G., Hespanha, J., Shahabi, C., Ortega, A., & Medioni, G "The Haptic Museum," Proc. EVA 2000, Conference on Electronic Imaging and the Visual Arts, 2000.

[7] McLaughlin, M. L., G. Sukhatme, J. Hespanha, C. Shahabi, and A. Ortega, "Touch in Immersive Environments," Proc. EVA 2000 Conference on Electronic Imaging and the Visual Arts, Edinburgh, Scotland, 2000.

Haptic Control of a Simplified Human Model with Multibody Dynamics

James J. Troy
The Boeing Company
jim.troy@boeing.com

ABSTRACT

This paper describes a haptically enabled human model controlled by multibody dynamics. The application implements a reduced degree of freedom dynamics model, which is a prototype for a system with a higher number of degrees of freedom. This work is meant to provide insight into the response of a constrained multibody dynamics system to real-time haptic interaction. The model is manipulated through virtual coupling attachment points, allowing the user to feel inertia and environment contact reactions through a force feedback device. A brief comparison is made between 3-DOF and 6-DOF haptic devices regarding depth of interaction achieved for this application.

1. INTRODUCTION

Traditional human modeling user interface tools can be difficult to use for complex posture adjustments, especially in confined spaces where contact with multiple objects is required. A more efficient interface that includes arm contact force feedback should be able to take advantage of insights the user already has about getting an arm into and out of confined configurations. Achieving this type of interaction requires developing a physically based model of the system and integrating it with a haptic (force feedback) interface device.

Purpose of this paper is to introduce a prototype articulated human model that demonstrates the type of interaction possible with multibody dynamics, interactively controlled by a haptic device. The implementation described here is an initial design with three degree of freedom (DOF) limbs constrained to move in a plane.

Motivation: The ultimate goal of this line of research is build a fully functional virtual environment for simulation. A big part of that is a human model that can interact realistically with objects in a virtual world. In an engineering environment, the primary use for a haptically enabled human modeling system is in the area of accessibility analysis for manufacturing and maintenance. This type of simulation environment would offer designers and analysts a more efficient way to answer basic ergonomics questions like “can a real mechanic get his/her arm in there when removing that part?”

The ability to interact with the environment without a complete graphical representation of the scene is an important aspect of a fully functional simulation system. Although this may seem like a minor consideration, real-life maintenance and assembly applications often have situations where the mechanic cannot view all aspects of a task. In a haptic simulation environment, the level of interaction should be deep enough that users can work effectively even with visual obstructions.

2. BACKGROUND AND PREVIOUS WORK

With a few exceptions, most human modeling software uses kinematically defined posture control. Inverse kinematics and complex interpolation schemes have been developed to give the user a very detailed level of control over every aspect of the figure’s motion. Most of this type of work had its start in robotics research [2], and has been adapted to work with computer graphics animation. In order to make something look like it is responding to external influences, animators need to make a lot of kinematic adjustments and use special spline functions to make the motion look realistic. Using kinematics alone as the driver for a haptic simulation is very difficult.

Although inverse kinematics solutions for the arm can keep up with the motion of haptic position inputs without causing too much of a strain on performance, there is no direct way to produce reaction forces from a kinematic solution alone. Collision response, inertia, momentum, compliance, and gravity are some of the aspects of a realistic simulation that are not part of a kinematics-based solution[1]. Heuristics can be defined to simulate some types of forces, but these are special case solutions that tend to be incomplete and cumbersome to maintain.

Dynamics offers a better general purpose solution. In a physically based system, Newtonian mechanics are used as the basis for motion. This is a better alternative for haptics, since forces are already part of the calculation. Efficient formulation of the dynamic equations of motion allow simulation to take place at interactive rates.

Recently, interactive control of physically based human model simulation with non-force feedback input has been accomplished [7][4]. But this level of physically based, interactive control for articulated figures has not been fully explored with haptics as the interface mechanism.

In addition to an efficient dynamics model, another key aspect of physically based haptics simulation is an efficient collision detection and force generation algorithm. Collision forces must be computed at high rates (1000Hz) in order to maintain simulation and haptic device stability. Achieving high rates in increasingly complex environments is an active research subject [3][5].

1. In the strict definition, kinematics is the study of motion without regard to the forces required to achieve it.

3. PHYSICALLY BASED MODELING

Developing a motion generation procedure based on the laws of physics will be the focus of this section. The basic process involves four steps: 1. derive efficient equations of motion for a system of interconnected bodies, 2. include control forces and torques, 3. include collision forces and torques, and 4. solve the equations of motion using numerical integration methods.

3.1 Multibody Dynamics

The first tasks in deriving the equations of motion will be to set the scope of problem and to define the structure of the multibody system.

Since most human modeling applications in engineering design analysis involve situations in which arm motion is the primary focus, derivation of the dynamic equations for the arm will be the main topic of this discussion. A full human arm has seven articulated degrees-of-freedom[2], but for this application a simplified 3-DOF articulated arm model will be implemented with motion constrained to the sagittal plane (the motion plane seen from a side view).

In order to reduce the computational effort required to obtain a solution, generalized coordinates will be used to define the equations of motion. Generalized coordinates describe the system with the minimum number of equations necessary; one independent variable will be solved for each degree of freedom. In general, the equations will be more complex to derive than those described in Cartesian coordinates, but there will be far fewer for the computer to solve at run time. Figure 1 shows the generalized coordinates and dimensions for the 3-DOF arm.

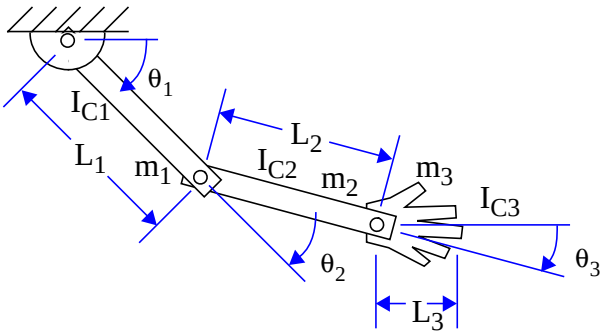


Figure 1. Planar 3-DOF arm coordinates and dimensions

The equations of motion will be second order ordinary differential equations (ODEs). For this application, the equations are derived by using the Lagrange method [8], which is a technique for describing the energy exchange between kinetic and potential forms. The Lagrange method begins by defining the motion of each body segment in terms of the partial differential equation described in Equation 1.

2. An unconstrained arm has seven primary degrees of freedom, not counting individual finger motions or shoulder translations.

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} = Q_i \qquad (1)$$

Where q_i is the generalized coordinate, T is the scalar kinetic energy equation, V is the scalar potential energy equation, and Q is the virtual work. The resulting nonlinear equations are in the form, $A\ddot{X} = B$ and are then solved for in terms of the accelerations, $\ddot{X} = A^{-1}B$. Where $\ddot{X}$ is the vector of accelerations, A is a symmetric matrix of mass and inertia terms, and B is a function of the generalized velocities, positions, control forces, and collision reaction forces. Since the generalized coordinates are joint angles, all of the forces due to collisions and control inputs are converted into moments and appear in the B vector.

3.2 Control

For a haptics application, forces and torques transmitted to the end effector will be necessary.

Control of the arm is accomplished though a virtual spring/damper coupling attached to the wrist, as shown in Figure 2. This acts like a proportional-derivative (PD) controller [6], and will need to be tuned depending on the mass and inertial properties of the system. Other solution methods, like joint space control, can offer better goal position following, but requires an inverse kinematics calculation which adds to the computational overhead.

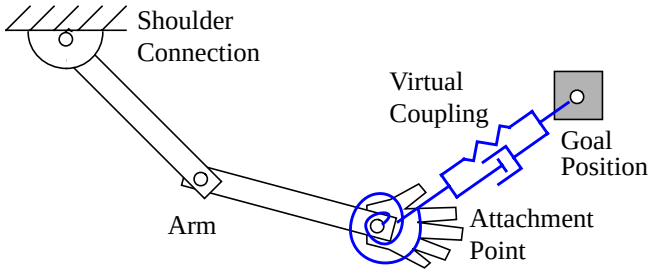


Figure 2. Arm with virtual coupling element

The equations for the control forces and torques transmitted to and from the haptics end effector are listed below.

$$U = -(X_{goal} - X_{attach})K_p - (\dot{X}_{goal} - \dot{X}_{attach})K_d \qquad (2)$$

Where $U = [Fx,Fy,Fz,Tx,Ty,Tz]^T$, and K_p and K_d are the proportional and derivative gains, respectively. Note that Fx, Tx, and Ty are not part of the equations of motion for this 3-DOF arm. They can be removed from the equation above and used as a control mechanism for other aspects of the simulation. The forces and torques are described in terms of the wrist position (or ankle for a similarly configured 3-DOF leg), are then sent to the haptic device. For the PHANTOM®, the calculateForceFieldForce GHOST® functions are used to accomplish this.

3.3 Collision Detection

A simple polygonal-based system was initially developed to generate the appropriate reaction forces, but was eventually

replaced with a more efficient force generation method that allows the object sizes and complexity to be scaled effectively.

The system currently uses a voxel based collision detection and contact force generation algorithm called Voxmap PointShell™ (VPS) [5]. This method defines each object as a collection of voxels and surface points. When the relative motion of two objects places a surface point of one object in the same volume as a voxel of the other, a contact event is detected and a penalty based force is generated.

Due to the voxelization process, the forces generated by VPS tend to be somewhat jaggy when just a few of the surface points are in contact, but this effect diminished when more of the points are in contact. The accuracy of collision position and force generation for this method is based on the size of the voxels and the number of surface points. If more memory and processing power are available then the size of the voxels can be reduced and the number of surface points increased. On the other hand, if more processing power is available, then a more complex dynamic model could be implemented instead.

One of the key challenges here is to balance the complexity of multibody dynamics computations with the accuracy of the voxel based collision detection method. For this application the system was tuned so that the amount of time spent in integrating the equations of motion is approximately the same amount as that spent generating collision forces.

Collecting Collision Results

In this application, collisions are processed in pairs instead of simultaneously. Objects with relative motion are fed two at a time into the collision detection algorithm and the forces from all collisions are collected and included in the equations of motion. The total number of pairs to be checked at each update is the combination:

$$C(n, 2) = n(n-1)/2 \quad (3)$$

For a system which includes a single three link arm and three external objects, the total number of pairs to be checked at each update is 15. As the system becomes more complex this can quickly get out of hand. For example, a system with 16 moving objects and links (which is reasonable for a fully articulated human model) the total would be 120 pairs!

Culling the list is important to achieving usable performance for larger systems. Taking advantage of joint limits to reduce the number of potential collisions between segments is a first step. Predictive algorithms, temporal coherence, and spatial partitioning can also help to reduce the number by helping to decided what needs to be checked and what does not. An application should also be prepared to experience the worst case scenario with all, or many, of the components coming into contact at the same time. Since the complexity of the current 3-DOF arm application is relatively low, a higher level part culling algorithm was not implemented. This means that the worst case collision situation is always in effect.

3.4 Solving the Equations of Motion

In order to solve the decoupled equations of motion, which are described in terms of the accelerations, $\ddot{X}$, the equations will need to converted into a series of *2N* first order equations in preparation for numerical integration. To solve the first order system, a constant time step numerical integration method is needed (a Runge-Kutta 4^{th} order method was used here). The time step is adjusted so that the simulation can maintain stability in the case where a large numbers of collisions occur simultaneously. Adaptive time step methods are not used since the speeding up and slowing down of these methods causes inconsistent performance.

3.5 Other Implementation Issues

To increase application usability, a simple first order translation function can be used to pull the figure around the environment. It activates when an arm is fully extended and wrist forces reach a specified value. The only variables to adjust here are the activation force value and a velocity gain, which is a linear function of the wrist force.

Since the user has only one haptic device in this application for controlling multiple limbs, only one arm or leg can be independently controlled at any one moment in time. The active limb is controlled through attachment points on the wrists or ankles. When an arm is not actively controlled by the haptic device and multibody dynamics model, it stays in a locked posture at the previous position. An inverse kinematic function automatically positions inactive legs to keep the feet on the ground.

A limitation of this system is that the limbs are not dynamically coupled to each other through the torso. In a simulation environment with a single haptic device, simultaneous operation of both arms is not possible, so this limitation is not critical here. Dynamic coupling will become more important when two handed haptic applications are implemented or full body interaction is required.

A momentum transfer step is required when picking up objects. This is treated as an inelastic collision. Mass and inertia of the combined hand and object segment must be recalculated. The collision pairs list will also need to be updated to avoid calculating unnecessary collisions for the reconfigured system. It should be possible to pick up several objects using the same process for each new object, although this is not currently implemented.

As mentioned earlier, the complexity of the system model is limited by the processing power available. Since the haptic device must be updated at 1000Hz, it is best to try to match this rate with the model dynamics update rate. If the required update rate is not achieved, the forces sent to the haptics device are kept the previous values, and the computation falls into the next haptic refresh cycle. The result of unmatched updates is usually an unsatisfying washboard-like force effect.

4. RESULTS

A collection of functional test environments were assembled to evaluate the interaction capabilities of the human model with a static scene and movable objects. These consisted of part extraction and environment interaction tasks. One of the test cases, in which a hand tool is used to interact with the environment, is shown in Figure 3. The application was evaluated on two 6-DOF haptic devices: the PHANTOM 1.5/6DOF and 3.0/6DOF; as well as a 3-DOF PHANTOM Desktop. The application was hosted on a 250MHz dual processor SGI Octane.

Figure 3. Dynamics application with 6-DOF PHANTOM

Initial development was done with the 3-DOF PHANTOM Desktop device (which measures all six degrees of freedom, but only has force output on the three translational axes). Collisions with all parts of the arm and tool are transmitted to the users hand. Inertia and gravitational forces of the arm are also transmitted through the wrist of the model to the user's hand.

Attempting to extend the arm past the limit of reach produces a restoring force. If extended further, the force increases until it reaches a specified limit, after which the figure is translated along the direction of the force. This allows the user to drag the body to a desired location and seems to be a very natural way to interact with the model.

Although the PHANTOM Desktop haptic device does not have the ability to output torques at the end effector, the dynamics of the arm model allows some indirect artifacts of rotational motion to be perceived by the user. Translational motion is generated through the coupling of the wrist and arm segments, which provides important cues that wrist rotations are affecting the system. When simultaneously viewing the model motion on the screen and feeling the translational output it is possible to train yourself to accept this type of reaction as a partial substitute for true rotational force output.

When the same application was applied to a haptic device with 6-DOF force feedback the wrist torques were directly available for output. This gives the system a more natural quality feeling that comes closer to the goal of being able to explore the virtual environment without looking at the screen. As would be expected, the torque reaction is amplified by using extended hand tools like a hammer, pipe wrench, or tennis racket. The higher torque output of the larger of the two 6-DOF haptic devices (the PHANTOM 3.0/6DOF) gives a more convincing reaction in these amplified cases. After using the application on a device with 6-DOF force feedback, and feeling the interaction of the wrist torque due to collisions and rotational inertia of the various tools, going back to using the 3-DOF feedback device gives the user a sense that something is missing — or not working properly.

These comparisons are subjective in nature. In order to draw more objective conclusions with respect to performance benefits of 6-DOF vs. 3-DOF haptics, more formal studies will be needed.

5. CONCLUSIONS

An articulated human figure defined by a multibody dynamics model was presented in which limb motions are interactively controlled by a haptic device. The primary contribution of this type of system for human figure manipulation in a virtual environment is in its ability to allow more natural interaction modes. The results show that a human modeling application, enabled by interactive multibody dynamics and force feedback, can be made to respond with a heightened sense of realism and functionality, especially when using a 6-DOF haptic device.

REFERENCES

[1] Baraff, D., "Fast Contact Force Computation for Nonpenetrating Rigid Bodies." Proc. ACM SIGGRAPH 94 Conf., Orlando, FL, pp. 23-42, July, 1994.

[2] Craig, J.J, *Introduction to Robotics, Mechanics and Control*, Addison-Wesley, Reading, MA, 1986.

[3] Gregory, A., Lin, M.C., Gottschalk, S., Taylor, R., "A Framework for Fast and Accurate Collision Detection for Haptic Interaction." In the Proc. of IEEE Virtual Reality Conf., Houston, TX, Mar., 1999.

[4] Laszlo, J., van de Panne, M., Fiume, E., "Interactive Control for Physically Based Animation." Proc. ACM SIGGRAPH 2000 Conf., New Orleans, LA, pp. 201-208, July, 2000.

[5] McNeely, W.A., Puterbaugh, K.D., Troy, J.J., "Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling." Proc. ACM SIGGRAPH 99 Conf., Los Angeles, CA, pp. 401-408, Aug., 1999.

[6] Ogata, K., *Modern Control Engineering*, 2nd ed, Prentice-Hall, Englewood Cliffs, NJ, 1990.

[7] Troy, J.J., "Real-Time Dynamic Balancing and Walking Control of a 7-Link Planar Biped." Proc. ASME Design Engineering Technical Conf., Atlanta, GA, Sept., 1998.

[8] Wells, D.A., *Theory an Problems of Lagrangian Dynamics*, McGraw-Hill, New York, 1967.

Haptically perceived orientation of a planar surface is altered by tangential forces

W. L. Sachtler bensan@mit.edu **Misty R. Pendexter** mistyp@mit.edu **James Biggs** jbiggs@mit.edu **Mandayam A. Srinivasan** srini@mit.edu

Laboratory for Human and Machine Haptics
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

We describe an experiment showing that addition of tangential forces can alter the haptically perceived orientation of a flat surface in a systematic fashion. Using a stylus mounted on a Phantom controlled by the GHOST software package, subjects performed single horizontal strokes along vertical surfaces at a number of orientations around the mid-sagittal plane. After each trial, subjects (who could not view their hand) judged whether the surface was rotated around the vertical axis in the clockwise or counter-clockwise direction from the sagittal plane. Psychometric curves for perceived surface orientation were measured in this way for various magnitudes of tangential forces along the stroke direction or in opposition to it. Psychometric curves shifted systematically with varying force magnitude, indicating that resistive forces tilted the perceived surface into the path of motion, while assistive forces tilted it away. More generally, the results show that perception of global surface orientation is not exclusively based on the location of the surface in space, but also on the forces encountered along the surface.

Introduction

In order to haptically determine surface shape through a stylus or similar tool, one could envision a method of sampling contact points in space and performing some kind of interpolation between them. However, there is evidence suggesting that, in humans, the forces encountered during a haptic scan affect the shape percept, suggesting a richer set of perceptual mechanisms. For example, as mentioned in Minsky (1995), lateral assistive forces on a haptic interface create the percept of moving downhill. Morgenbesser & Srinivasan (1996) demonstrated that particular modifications of force vectors on a nominally flat surface ("force shading") were perceived as a bump.

It is unclear, however, whether we have simply learned to associate certain changes in resistance with particular shape features. For example, an increase in resistance followed by a decrease would generally be associated with a bump.

In the experiment described here, we sought to determine whether tangential forces can in fact modify the global perceived orientation of a surface. This was done by measuring the perceived change in orientation of a flat surface as a function of the magnitude of an added tangential force. Specifically, we tested a range of orientations under a number of force conditions, and from the results computed an estimate of the perceived sagittal plane for each condition. This is a form of nulling technique to cancel the perceived change in orientation with a real change in the orientation of the surface.

Estimating the magnitude of a perceptual phenomenon by nulling it with variations in the physical stimulus has been applied to a number of perceptual phenomena in the past (Taylor, 1963; McCourt, 1982; Krauskopf et al, 1986; Sachtler & Zaidi, 1993).

Equipment

Experiments were performed using a Phantom 1.0x fitted with 4000-counts-per-revolution encoders and a standard stylus. Surfaces and tangential force fields were generated using the GHOST software package running on a dual-500 MHz-processor Pentium PC under the Windows NT 4.0 operating system.

Stimuli & Procedure

Stimuli consisted of flat vertical surfaces with a spring constant of 0.8 N/mm and no friction.

Surfaces were presented at a range of orientations, in increments of 6 degrees, around the mid-sagittal plane. A value of zero corresponds to a surface aligned with the mid-sagittal plane. Figure 1 shows a top-down view of a subject's relation to the stimulus. The two subjects whose results are reported here were both right-handed. They were instructed to stroke the surface with the stylus by always moving their right hand towards their body.

Force fields tangential to the surfaces were added along the horizontal direction, that is, parallel to the ground, and were directed either towards or away from the subject. Figure 2 summarizes the sign conventions for force fields at positive and negative surface orientations.

Positive forces were directed towards the subject, assisting motion of the stylus tip along the simulated surface, while negative forces were directed away and thus resisted movement of the stylus.

Seven force conditions were tested with magnitudes of –0.9, -0.6, -0.3, 0.0, 0.3, 0.6, and 0.9 Newton (N). Tangential forces were applied only when the endpoint of the stylus was in contact with the simulated surface.

Figure 3 summarizes the procedure for a single trial, as described in the caption. Subjects, who could not view their hand, were seated at a desk, resting their right elbow on its surface, and used a chin-and-forehead rest to maintain a fixed location with respect to the Phantom throughout the experiment. They were instructed to draw the tip of the stylus along a horizontal path on the vertical surface. Movement and forces were restricted to the horizontal to minimize confounds with gravity. At the end of each trial, subjects indicated via key press with their other hand whether they judged the right side of the surface to be oriented towards or away from them, that is, whether the surface was rotated in the clockwise or counter-clockwise direction from the sagittal plane. They were instructed to guess when unsure.

Seven surface orientations for each of the seven force conditions were randomly interleaved for a total of 49 trials in one block. Eight such individually randomized blocks were run for each subject, providing psychometric curves for perceived surface slant.

The midpoint of each psychometric curve--with equal proportions of clockwise and counter-clockwise orientation judgments--served as an estimate of the perceived sagittal plane. That is, the angle at which a surface was perceived as not tilted. The range of orientations tested for each

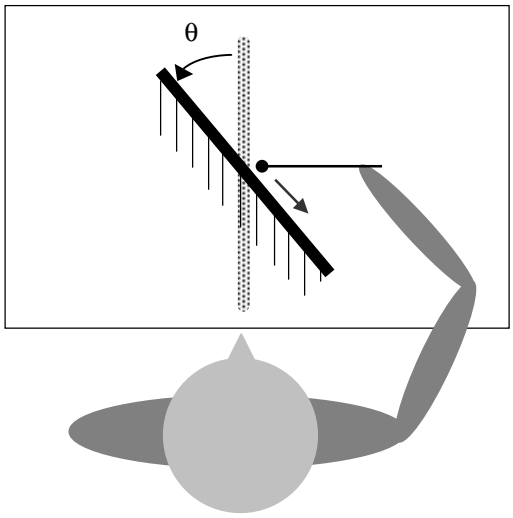


Figure 1: Top-down schematic view of a subject performing a single haptic scan of a simulated surface (solid line) with a Phantom stylus. Surface orientation (θ) is defined with respect to mid-sagittal plane.

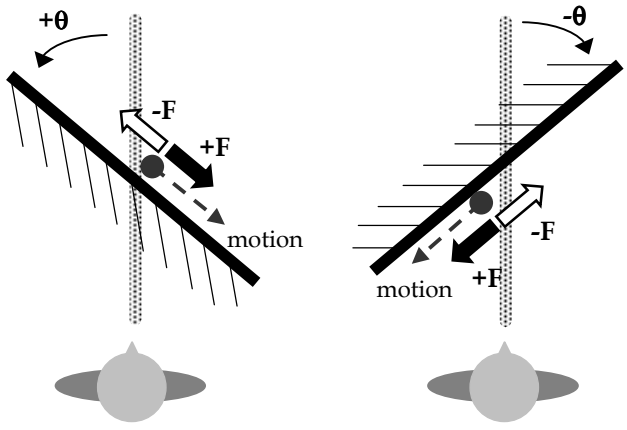


Figure 2: Sign conventions (top-down view). Positive tangent forces are directed towards subject, in alignment with motion of stylus tip along a surface. Positive angles correspond to counterclockwise rotation of the surface with respect to mid-sagittal plane.

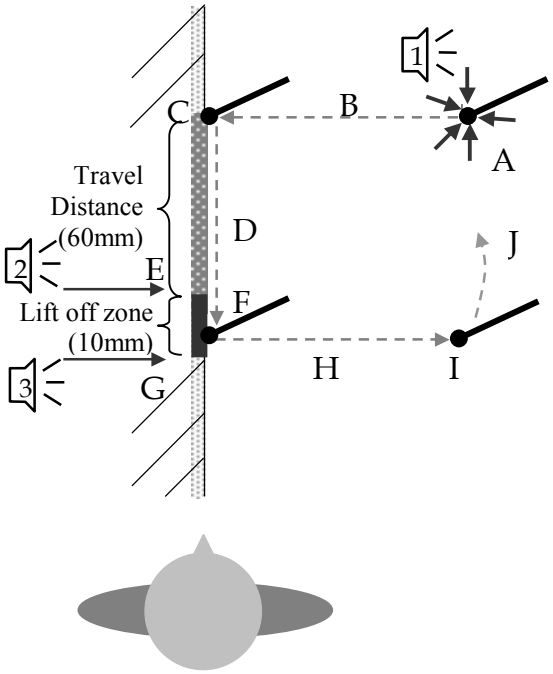


Figure 3: Procedure

A) All trials began from the same starting point. Phantom was constrained under software control.
B) A tone (1) indicated when the constraint was released, and subjects moved the stylus towards the virtual surface.
C) Point of first contact on surface served as reference for distance traveled along the surface.
D) Subjects drew stylus towards their body along the surface.
E) A second tone indicated when desired travel distance from first contact was reached (60mm).
F) Subjects were required to lift stylus off surface within 10mm after travel distance limit or trial was cancelled.
G) A third tone indicated if passed beyond liftoff zone to provide feedback.
H) Subjects lifted stylus off the surface.
I) Subjects responded via key press with free hand while holding stylus in space.
J) Once the response was recorded, the Phantom was pulled to the starting point under software control, and a new trial began.

force field was determined in pilot runs.

Two right-handed subjects were tested. DHP was naïve regarding the purpose of the experiment.

Results

Results for each subject are shown in separate columns in Figure 4. Panels show psychometric curves for perceived surface slant at different force field magnitudes indicated on the left.

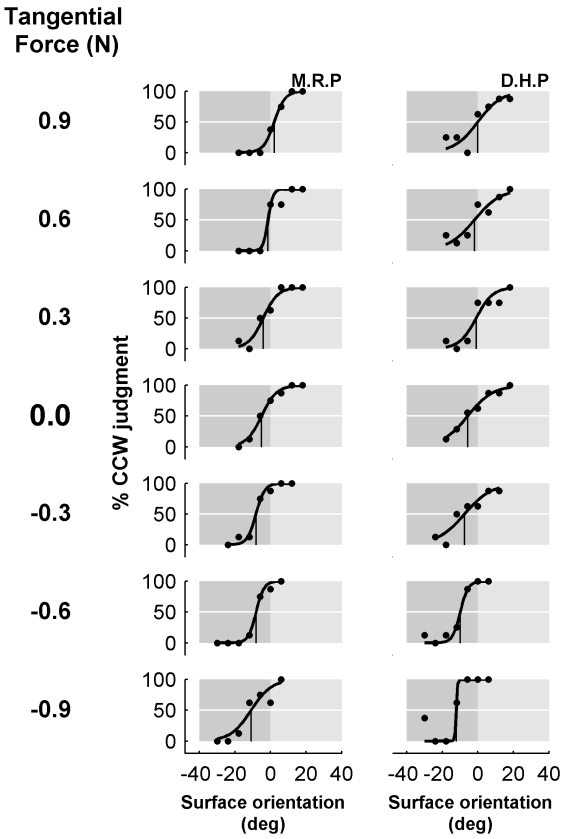


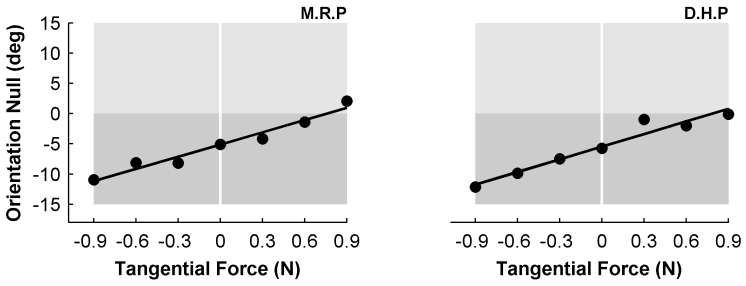
Figure 4: Results for two subjects are shown in separate columns. Each panel shows psychometric curves for perceived surface orientation for different tangential force fields, indicated on the far left. Surface orientation is plotted on the abscissa, while the ordinate shows the proportion of trials in which subjects indicated the surface was rotated counterclockwise (CCW) with respect to the mid-sagittal plane. Solid curve shows the best-fitting logistic function used to estimate the angle at which clockwise and counterclockwise judgments were balanced.

Surface orientation is plotted along the abscissa, while the ordinate of each panel indicates the percentage of counter-clockwise judgments (thus, response percentages are high for positive--counter-clockwise--rotations). A value of 50% indicates that the proportion of clockwise and counter-clockwise judgments was the same.

A logistic curve, shown as a solid line, was fit to each data set to estimate the surface orientation at the 50% mark. A thin vertical line in each panel indicates this value, which served as a measure of the angle at which a surface did not appear tilted.

The position of the psychometric curves varied systematically with the applied force field. This is

Figure 5: Orientation of a surface that was perceived as sagittal (not tilted) as a function of tangential force field magnitude. The straight line in each plot shows a linear regression fit, with slopes of 6.7 and 7.0 for MRP and DHP, respectively. Correlation coefficients were 0.98 and 0.97, respectively



shown in different form in Figure 5, where the orientation of a surface that was perceived as sagittal is plotted as a function of force magnitude. That is, applying a particular tangential force to a tilted surface led subjects to report that it was not tilted.

There was a negative bias for both subjects in the perceived orientation of a sagittal surface even in the absence of a force field (bias MRP: -5.1 deg, DHP: -5.5 deg). The angle at which surfaces did not appear tilted increased for positive force fields, and decreased for negative ones.

Summary

Tangential forces systematically altered the haptically perceived orientation of a flat surface.

Results indicate that constant resistive forces tilted the perceived surface into the path of motion, while assistive forces tilted it away.

The linear relationship between tangential force and perceived surface orientation suggests a method by which to scale the lateral forces of 2D force-feedback devices in order to simulate surface features extending in three dimensions.

References

Krauskopf, J, Zaidi, Q, & Mandler, MB (1986). Mechanisms of simultaneous color induction. *Journal of the Optical Society of America,* A3, 1752-1757.

McCourt, ME (1982). A spatial frequency dependent grating-induction effect. *Vision Research,* 22, 119-134.

Minsky, MDR (1995). *Computational Haptics: The Sandpaper System for Synthesizing Texture for a Force-Feedback Display.* Ph.D. Thesis, Massachusetts Institute of Technology.

Morgenbesser, HB & Srinivasan, MA (1996). Force shading for haptic shape perception. *Proc. Fifth Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, ASME Dynamics Systems and Control Division, DSC-Vol. 58,* 407-412.

Sachtler, WL and Zaidi, Q (1993). The effect of spatial configuration on motion aftereffects. *Journal of the Optical Society of America,* **A10**. 1433-1449.

Taylor, MM (1963). Tracking the decay of the after-effect of seen rotary movement, *Vision Research* **7**, 427-439.

The Influences of Network Issues on Haptic Collaboration in Shared Virtual Environments

Soju Matsumoto† Ichiro Fukuda† Hironao Morino†
Kenji Hikichi† Kaoru Sezaki‡ Yasuhiko Yasuda†

† Graduate School of Science and Engineerng, Waseda University
matumoto, fukuda@yasuda.comm.waseda.ac.jp

‡ Institute of Industrial Science, University of Tokyo
sezaki@mcl.iis.u-tokyo.ac.jp

Abstract

There is currently much discussion of Quality of Service (QoS) measurements at the network level of real-time multimedia services.
The presence of a network brings up a number of issues. In case of visual or auditory interaction, the effect of such issues have been treated by many researchers and the effective countermeasure have been proposed. However, little is known about the relationship between network issues and networked haptic interfaces in virtual environments(VEs). Our researches aim at investigating and clarifying the above issue and constructing more robust haptic interaction system to realize network-based shared virtual environments(Net-SVEs) under realistic network conditions - from the end-user's point of view. For this purpose, we have designed some experiments to investigate the influence of network delay on multiuser haptic collaboration system through subjective and objective assessment by users. This paper shows the results of the relation between QoS and network delay on haptic Net-VEs.

1 Introduction

The number of real-time multimedia applications over network has been increasing steadily, and with it, to measure and assess Quality of Service (QoS) are becoming more and more important. There has been a surge in literture addressing QoS issues. But the emphasis has been on the QoS of conventional "one-directional" multimedia(i.e, visual and auditory. Farthermore, most of them focus on QoS issues at the network level, rather than from the end-user's point of view. Since it is the end-user who will determine whether a service or application is a success, it is vital to carry out subjective assessment of the multimedia quality delivered through these.

2 Network Issues on Haptic Interacition

Network issues that is, delay, jitter (variation of delay), reliability, or bandwidth may cause severe deterioration of stability or performance of the system. They are unavoidable in realistic network environments, so need to be discussed carefully. In this section, we discuss these issues focused on network latency and communication architecture.

2.1 Network Latency and Force Feedback

In networked remote haptic interaction in Virtual Environments(VEs), delay may cause not only time lag between human operation and force feedback, but system instability like an excessive rebound or vibration of reaction force. As regards impedance display(sensing motion and producing force)like PHANToM, interaction force between haptic device end point and virtual object is calculated based on Springdumper model. Reaction force is generated in proportion to the depth of a PHANToM cursor in a virtual object and the relative velocity between a cursor and a object. Without delay between haptic display and VEs , reaction force calculated in VEs is output to haptic device instantly, but when there is some delay, a time lag between generating and outputting reaction force occurs. For this time lag, haptic end point might penetrate deeper in a virtual object till it is pushed back by reaction force. By the time the force feedback arrives haptic display, the penetration have already increased, so large forces are generated. This effect is felt as a rebound or vibration of a haptic display by users.

Differences of delay time between each users in SVEs is also troublesome. Network states - including delay time - in each client are not the same in many cases.

This may cause inconsistency of collaboration.

2.2 Communication Architectures

Generally, we can choose various communication architectures to realize SVEs, for example, client-server or peer-to-peer, or a mixture of them and so on. They have different features respectively, so we need to choose more suitable ones considering of the overall system along with their advantages and disadvantages.

Suppose the simplest case of Peer-to-peer architecture. In this case, all clients have whole SVEs information independently, and required data to be transmitted is only PHANToM position information. In this architecture, to guarantee consistency of SVEs on condition that exists fixed delay, we only have to delay timing that local PHANToM position is displayed to the maximum delay time of all the peer-to-peer connection. This method provides absolute consistency for SVEs without excess penetrating. However, as an increase of a number of clients the connections become complex, besides each client requires a high machine power to calculate all SVEs . In the client-server architecture, a required machine power of each client is lower and connections are simpler than peer-to-peer. Moreover, by managing SVEs in a central server it can keep consistency of collaboration. But the differences of delay may cause inconsistency of collaboration.

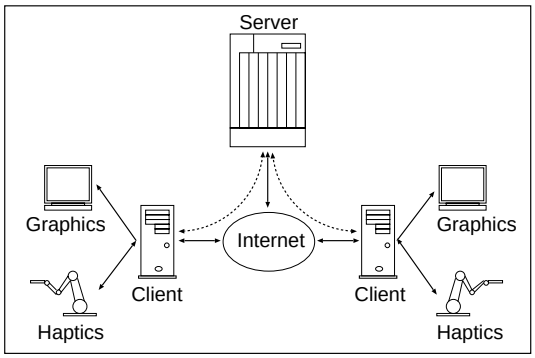


Figure 1: Client-Server Model of Net-SVEs

3 Design of System

In consideration of the issues described in previous section, we designed a prototype of network-based haptic collaboration system for subjective and objective assessment.

This time, we adopted Client-Server architecture for network-based haptic collaboration system. In this architecture, physical simulation or management of entity state are processed at a central server almost all together in order to keep consistency of collaboration. Details of each part of the system are provided below.

3.1 Control Design

To avoid generation of excess reaction force, it is calculated at each client. In Figure 2 , Fe, the force calculated in a server, is applied to VEs managed by the server, and Fm, the force calculated in clients, is applied to haptic display. A server transmits information of object surface state which is contacting to server PHANToM position, to each client. Each client calculates the reaction force based on interaction between the contact object state from the server, and client PHANToM position.

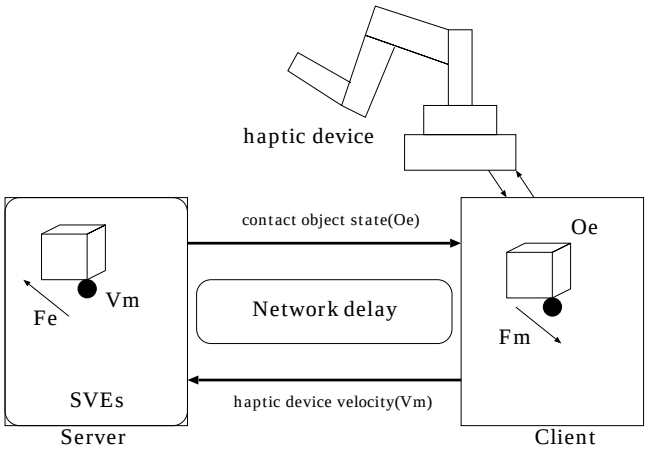


Figure 2: Control Design for haptic net-VEs

3.2 System Specification

- Server
  PLATFORM :Windows NT4.0
  CPU :PentiumIII 600MHz
  RAM :128MB SDRAM
  NIC(10/100BASE-TX):3COM 3C905B-J-TX
- Clients
  PLATFORM :Windows NT4.0
  CPU :PentiumIII 850MHz
  RAM :512MB SDRAM
  Display-Adapter:ELSA Gloria-XXL with 3D shutter glasses
  Force Device :PHANToM PREMIUM 3.0L
  NIC(10/100BASE-TX):3COM 3C905B-J-TX

4 Experiments

4.1 Experimental Overview

In the experiments, the users perform a task along with a particular rule in a VE. Network delay from

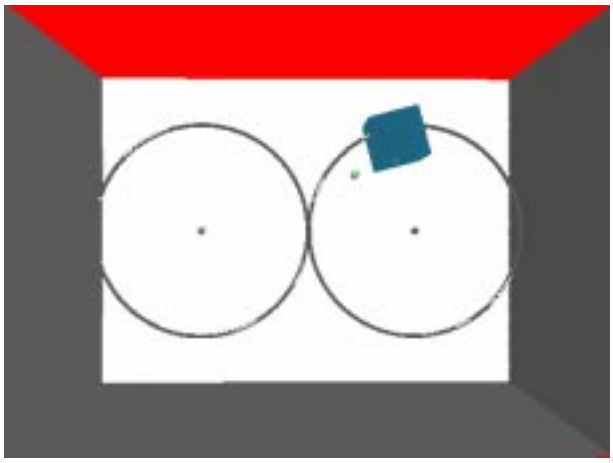


Figure 3: A View of the Experiments in SVEs : Users manipulate a dynamic cube in order that the moving target is always within the cube in collaboration. The target moves along the two circles at 30 seconds per cycle.

the server to each user is changed arbitrarily and independently, and the deterioration of performance is measured. We characterized the system performance as Quality of Service(QoS). QoS is measured by both subjective and objective(quantitative) assessment. Our experiment are designed for both single-user and multi-user system. At present, single-user experiment has always conducted.

4.2 Experimental Design

A task of both experiments is the same, except motion constraint of virtual dynamic object.

In VEs(Figure 3), there is a small moving target. Users manipulate a dynamic cube in order that the target is always within the cube. The target moves along the two circles at 30 seconds per cycle. We constrained the motion of the object on the 2-Dementional plane, to control the difficulty of this task.

4.2.1 Experiment1(single-user)

As preparation, we have experimented on networked single-user haptic operative system in a VE - without collaboration. This is aim to investigate the influence of delay on a sense of operation of virtual object in client-server architecture. The dynamic cube does not rotate in the single-user VE, so a user lifts up and moves the cube by supporting the bottom surface.

4.2.2 Experiment2(multi-user)

Based on the results of an experiment1, we implemented networked multi-user collaborative system in a SVE. This aims to investigate the influence of difference of delay between each client on a sense of accomplishment of a task. In case of multi-player system, the cube behaves as rigidbody, so users must nip and manipulate cube in collaboration as keeping the balance.

4.3 Assessment Methodology

4.3.1 Subjective Assessment

Since there are no recommendation of haptic quality assessment methodology, we adopted the method of assessing quality of image[1]. In this case, comparisons to reference conditions (i.e.no network delay) are made using the double-stimulus continuous quality scale (DSCQS) and the double stimulus impairment scale (DSIS). The scale of both methods, 5-point quality scale and impairment scale are defined at Table1.

A Subjective quality is assessed based on, "controlablity of the object"(using DSIS), "a feeling of touch to the object"(using DSCQS), and"a sense of fulfillment of the collaborative task"(using DSIS, multi-user only).

Table 1: 5-point opinion scale

Score	Impairment Scale	Quality Scale
5	Imperceptible	Excellent
4	Perceptible, but not annoying	Good
3	Slightly annoying	Fair
2	Annoying	Poor
1	Very annoying	Bad

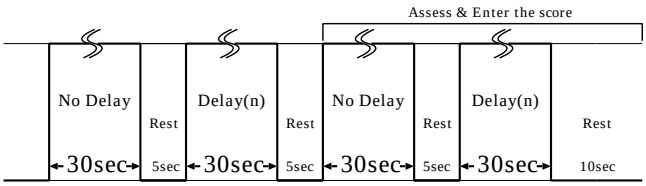


Figure 4: Presentation method of assessment

4.3.2 Objective Assessment

A quantitative(objective) measures for the "the performance of the task" was derived from how long time users could keep a moving target within the cube.

5 Results

The Experimental results are shown below. They are not reliable enough statistically, but indicate tendencies of the effects of delay on QoS in networked haptic interaction in a VE. Single-user experimental results are shown in Figure5-7. They indicate an allowable rate of delay are around 60ms in the system.

Conclusions

This paper reports the prototype experimental system design and implementation for investigating the influence of network delay on QoS of haptic interaction in SVEs, and shows the experimental results. The multi-user experiment is being conducted now, and the results will be shown in the near future.

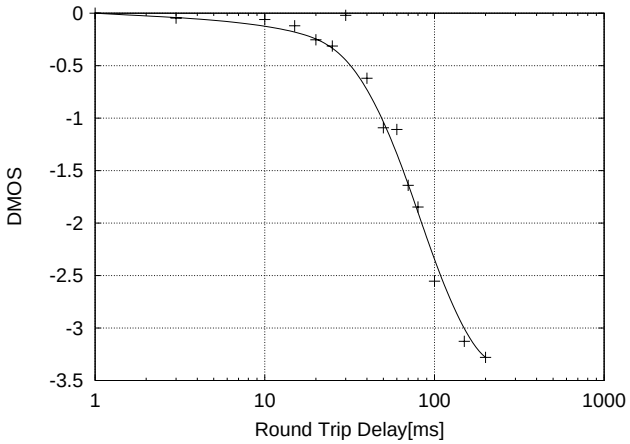


Figure 5: Controlability of the object – Delay(single-user)

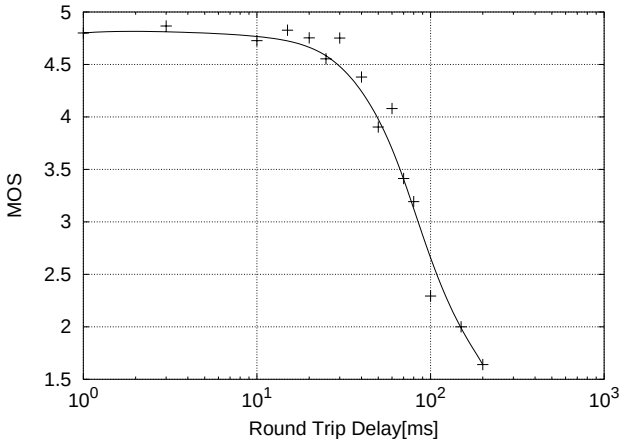


Figure 6: a feeling of touch to the object - Delay(single-user)

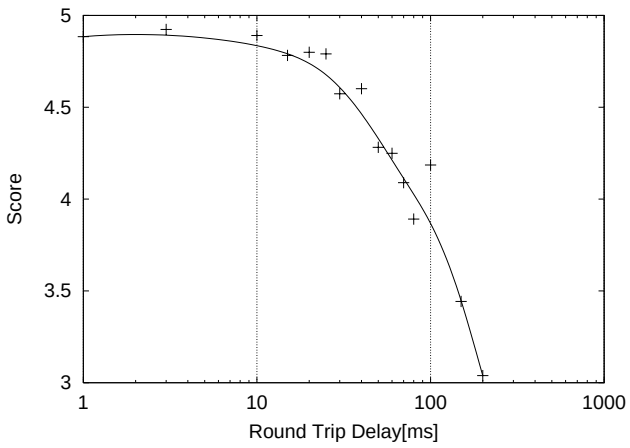


Figure 7: Performance of the task – Delay(single-user)

ACKNOWLDGEMENT

This work is partly supported by the Inamori Foundation.

References

[1] "Recommendation 500-8 ; Methodology for the subjective assessment of the quality of television pictures ", ITU-R Recommendations, 1998.

[2] T.Massie, K.Salisbury, "The PHANToM Haptic Interface: A Device for Probing Virtual Objects ", Procs of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, Chicago, IL, Nov.1994.

[3] Ho, C., Basdogan, C., Slater, M., Durlach, N., Srinivasan, M.A. (1998). An Experiment on The Influence of Haptic Communication on the Sense of Being Together, British Telecom Workshop on Presence in Shared Virtual Environments, Ipswich, June 10-11

[4] "A Constraint-based God-object Method For Haptic Display", Craig Zilles and Ken Salisbury, IROS '95.

[5] "Networked Virtual Environments Design and Implementation",Sandeep Singhal and Michael Zyda, Addison Wesley

LEM - An approach for physically based soft tissue simulation suitable for haptic interaction *

Remis BALANIUK
Center for Advanced Techn. in Surgery
Stanford University
remis@robotics.stanford.edu

Ivan F. COSTA
SHARP/GRAVIR
INRIA Rhône-Alpes
Ivan-Ferreira.Costa@inrialpes.fr

Abstract

This paper presents LEM - Long Elements Method, a new approach for physically based simulation of deformable objects, suitable for real time animation and haptic interaction. The method implements a static solution for elastic global deformations of objects filled with fluid based on the Pascal's principle and volume conservation. The physics of the objects are modeled using bulk variables: pressure, density, volume and stress. The volumes are discretised in long elements. This discretisation has two main advantages: the number of elements used to fill an object is one order of magnitude less than in a discretisation based on tetrahedric or cubic elements; the graphic and the haptic feedback can be directly derived from the elements, and no intermediate geometric representation is needed. The use of static instead of PDE equations avoids all the problems concerning numerical integration, ensuring stability for the simulation and for the haptic rendering.

1 Introduction

The method proposed in this paper was conceived for soft tissue real time simulation, particularly for surgical simulation. The priorities in this kind of application are: unrestricted multi-modal interactiveness, including interactive topological changes (cutting, suturing, removing material, etc), physically based behavior, volumetric modeling (homogeneous and non-homogeneous materials) and scalability (high accuracy when needed).

The approach is based on a static solution for elastic deformations of objects filled with uncompressible fluid, which is a good approximation for biological tissues. The volumes are discretised in a set of Long Elements (LE), and an equilibrium equation is defined for each element using bulk variables. The set of static equations, plus the Pascal's principle and the volume conservation, are used to define a system that is solved to find the object deformations and forces. Global and physically consistent deformations are obtained (Fig. 1).

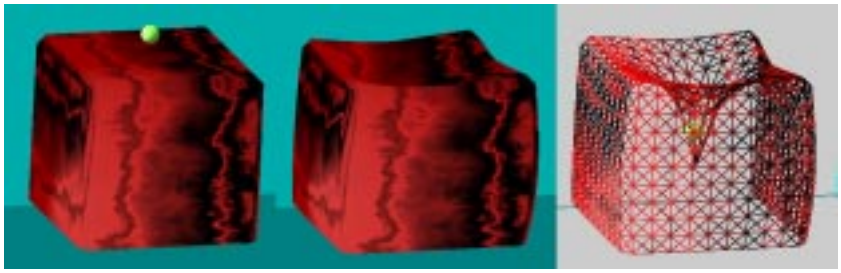


Figure 1: Soft-tissue touched by a rigid probe

For a survey of deformable modeling in computer graphics the reader is referred to [1]. Others recent methods proposed are the "Geometric Nonlinear finite element method" [3], the "Boundary Element Method" [2] and some medical simulators [4], [5], [6].

*Permanent address: (both authors) Universidade Católica de Brasilia - Brazil EPCT Q.S. 7, lote 1 - Águas Claras - 72030-170 - Taguatinga - DF - Brazil

2 Method Formulation

2.1 Pressure and Stress

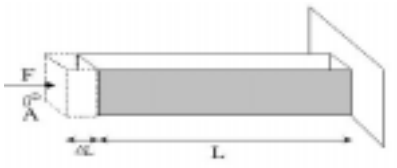


Figure 2: Long element

Consider the long elastic element illustrated in figure 2. The force F per unit of area A is defined as pressure: $P = F/A$. However the force per area unit producing the deformation is also the stress. For small applied forces, the stress s in a material is usually linearly related to its deformation (its change in length in our long elastic object). Defining elasticity E as the variable relating stress and the fractional change in length: $\Delta L/L$, it is possible to write: $s = E\Delta L/L$. Since the stress is related to the fractional change in length, the force can be related to the elongation ΔL in the well known form:

$$F = K\Delta L \qquad \text{where} \qquad K = AE/L. \tag{1}$$

Note that K is not constant, but it depends on the length L.

2.2 Static Solution

The static condition states that the forces, or pressures, in one sense have a correspondent of the same magnitude in the contrary sense on each point of the surface of the object , or: $P_{int} = P_{ext}$. The external pressure P_{ext} on the surface is affected by the atmospheric pressure and by the stress when an elongation exists, so:

$$P_{ext} = P_{atm} + E\Delta L/L. \tag{2}$$

The surface tension also affects the external pressure, as described further in section 2.4.

Considering that the object is filled by fluid, the internal pressure (P_{int}) is formed by the pressure of the fluid (without gravity) and the effect of the gravity acceleration (g), so:

$$P_{int} = P_{fluid} + dgh \tag{3}$$

where h is the distance between the upper part of the fluid and the point where the pressure is calculated. From the last three equations, a continuous equation can be obtained as:

$$E\Delta L/L - \Delta P = dgh \tag{4}$$

where $\Delta P = P_{fluid} - P_{atm}$.

Another external pressure to be considered comes from contacts between the object and its environment. At the points on the object surface, where are some external contacts, a term is added to the right side of equation 2. To obey the action-reaction law, the force applied to the external contact and to the object must to have the same magnitude. It means that the external pressure applied by the contact must be equal to ΔP. The elongation ΔL is defined by the penetration of the contact in order to make the surface follow the contact position (y). With these considerations, the equation 4 can be rewritten for the elements where there is external contact as:

$$\Delta L = y. \tag{5}$$

2.3 Long Elements

To simulate a deformable object we propose a discretisation of its volume in a set of long elements (Fig. 2). The idea is to fill the volume with long elements, to define equilibrium equations for each element based on the stated principles and to add global constraints in order to obtain a global physical behavior. A long element can be compared to a spring fixed in one extremity and having the other extremity attached to a point in the movable object surface. Different meshing strategies can be conceived to fill the objects. Applying the continuous equations (eqs. 4 and 5) for each of this elements we obtain:

$$E_i.\Delta L_i/L_i - \Delta P_i = d_i g_i.h_i \tag{6}$$

for the untouched elements. For the touched elements we obtain:

$$\Delta L_i = y_i \tag{7}$$

To make the connection between the elements two border conditions are applied:

1. Pascal's principle says that *an external pressure applied to a fluid confined within a closed container is transmitted undiminished throughout the entire fluid.* Mathematically:

$$\Delta P_i = \Delta P_j \text{ for any } i \text{ and } j. \tag{8}$$

   The first equation of this section (eq. 6) can then be written without the index i in the term ΔP_i.

2. The fluid is considered incompressible. It means that the volume conservation must be guaranteed when there is some external contact to the object. The volume dislocated by the contact will cause the dislocation of the entire surface, or in other words, the variation of volume due to the elements touched by the contact have to be equal to the sum of the volume created by the dislocation of all untouched elements to ensure the volume conservation:

$$\sum_{i=1}^{N} A_i \Delta L_i = 0 \tag{9}$$

   where N is the total number of elements.

2.4 Surface Tension

To reproduce the surface tension forces a number of terms will be added to the right side of the equation 2 corresponding to the neighborhood considered around the element. These terms are of the form $P = FA = kxA$, where x is the difference between the deformations of an element and its neighbor and k is a local spring constant. For a given element i the term relating its deformation to the deformation of its neighbor j is:

$$k_j(\Delta L_i - \Delta L_j)A_i \tag{10}$$

3 Mathematical Solution

Equations 6, 8 and 10 define the final equation for the untouched elements (considering 4 neighbors):

$$(E_i/L_i + 4kA)\Delta L_i - kA(\Delta L_{i-1} + \Delta L_{i+1} + \Delta L_j + \Delta L_l) - \Delta P = d_i g_i . h_i \tag{11}$$

where k and A were done constant for all elements to make easier the notation.

The untouched elements (equation 11) plus the elements in contact with the environment (equation 5) define a set of N equations, where N is the number of elements used to fill the object. Adding the equation of volume conservation (eq. 9) we have $N+1$ equations and $N+1$ unknowns: the pressure (ΔP) and the deformation of each element (ΔL_i for $i = 1$ to N).These $N+1$ equations can be written as a problem of the type $A.x = B$.

4 Method Implementation

The described method was used to implement a generic soft tissue VR simulator. The simulator was implemented in C++ in a Windows NT platform. This first prototype simulates deformations of a compliant object contacted by a rigid probe.

4.1 System organization

The system is organized around three decoupled main loops, executed concurrently in different processing units (threads, process and/or machines). The first loop simulates the deformations, the second renders the graphics and the third renders the haptics. The main loops share the data structure containing the long elements.

The objects are discretised in Cartesian meshes, each mesh containing long elements parallel to one axis of the reference frame. A Cartesian mesh defines a grid of parallel elements crossing the object. Each element starts in a point of the surface and crosses the volume until the end of the material, defining a line segment parallel to one of the reference frame axis.

Simulation loop The iterative biconjugate gradient method [7] is used to solve the system of equations defined in section 3. The static equations system does not demand any particular concern about time steps, stiffness or stability. The matrix is dynamically defined and the system $A.x = B$ can be rapidly solved. The solution (x) is the surface deformation, defined by a set of length differences in each element, and the difference in pressure.

Graphic loop OpenGL and GLUT are used to render the 3D volumes. There is no explicit geometric model of the object surface. In order to draw the object we use vertices directly derived from the long elements extremities and polygons defined between neighbor elements.

Haptic loop The LE representation of a volume is excellent for haptic rendering. The one point collision detection between the haptic probe position and the volume can be easily done using directly the LE cartesian meshes. Each mesh defines a grid, or a space filling map, and the collision detection in one mesh consists in checking the grid position corresponding to the probe position to see if the probe is penetrating a LE. Each mesh being parallel to one axis of the reference frame, the force feedback estimation is naturally decomposed. Each component of the force vector is independently estimated using the corresponding LE mesh.

During the collision two forces are been applied by the object to the haptic probe: a force applied by the touched elements (eq. 1) on the direction of the element and a force applied by the fluid inside the object (eq. 12). Multiplying both sides of equation 4 by the contact area A_c and comparing to equations 1 we obtain:

$$F = \Delta P.A_c + dgh.A_c \tag{12}$$

This force is perpendicular to the object surface and depends on the internal pressure, not on the penetration.

4.2 Results

In a standard dual 700MHz PC one iteration of the simulation loop takes about 0.05 seconds for a 600 elements mesh. The haptic interface was implemented using a PHANTOM haptic device (http://www.sensable.com). See figs. 1 and 3 for some examples of deformation. The global deformations are physically consistent and important phenomena such as the movement of all parts of the solid due to the preservation of volume are automatically produced.

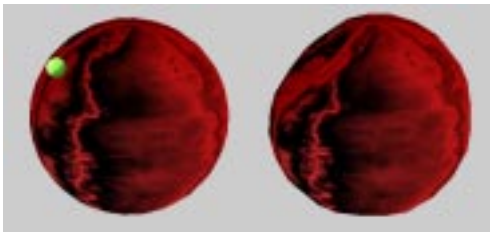


Figure 3: Soft sphere touched by a rigid probe

5 Conclusions

Utilizing the LE method have been able to physically model elastic deformations in a way that preserves volume, permits real time topology changes and is rapidly computable. The discretisation adopted by the method has two main advantages: the number of elements used to fill an object is one order of magnitude less than in a discretisation based on tetrahedric or cubic elements; the graphic and the haptic feedback can be directly derived from the elements, and no intermediate geometric representation is needed. The use of static instead of PDE equations avoids all the problems concerning numerical integration, ensuring stability for the simulation. No pre-calculations or condensations are used, in order to enable real time topology changes.

6 Acknowledgments

The authors would like to acknowledge the Cisco Corporation and SunMicrosystems for their support of this research. We would also like to thank Dr. Kenneth Salisbury, Dr. Thomas Krummel and the CATSS Laboratory at the Stanford University Medical School Department of Surgery; Christian Laugier and the INRIA; and the Catholic University of Brasilia for providing support, encouragment, and facilities.

References

[1] Saraj F. F. Gibson and Brian Mirtich. "A survey of deformable models in computer graphics." *Technical Report TR-97-19*, Mitsubishi Electric Research Laboratories, Cambridge, MA, November 1997 (http://www.merl.com/reports/TR97-19/index.html).

[2] D. James and D. Pai, "Artdefo Accurate Real Time Deformable Objects", *Computer Graphics*, vol. 33, pp: 65–72, 1999.

[3] Yan Zhuang and John Canny. "Haptic Interaction with Global Deformations". *Proceedings of the IEEE International Conference on Robotics and Automation - ICRA 2000*, 2428-2433, 2000.

[4] Murat Cenk Cavusoglu, Frank Tendick, Micahel Cohn and S. Shankar Sastry. "A Lapraroscopic Telesurgical Workstation". *IEEE Transactions on Robotics and Automation*, 15(4), 728-739, 1999.

[5] V. Vuskovic, M. Kauer, G. Szekely and M. Reidy. "Realistic Force Feedback for virtual Reality Based Diagnostic Surgery Simulators", *Proceedings of the IEEE International Conference on Robotics and Automation - ICRA 2000*, 1592-1598 (2000).

[6] Stephane Cotin, Herve Delingette and Nicholas Ayache. "Real-time Elastic Deformations of Soft Tissues for Surgery Simulation". *IEEE Transaction on Visualization an Computer Graphics*, 5(1), 62-73, 1999.

[7] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling. *Numerical Recipes in C.* Cambridge Univ. Press, 2 ed., 1992.

A Parameter Space for Perceptually Stable Haptic Texture Rendering *

Seungmoon Choi and Hong Z. Tan
Haptic Interface Research Laboratory, Purdue University
1285, Electrical Engineering Building
West Lafayette, IN, 47907-1285
{chois, hongtan}@ecn.purdue.edu

1 Introduction

After Minsky's pioneering work on synthetic texture rendering using a two-dimensional force feedback joystick [5], the research on haptic texture rendering has received much attention in the recent years, with a majority of work focusing on the development of texture rendering algorithms. Several successful implementations of texture rendering methods using various texture geometry models have been reported so far (for examples, see [1][2][4][6]). These studies focus on the computational aspects of texture rendering. Our research program is more concerned with the perceptual aspects, in additional to the computational aspects of texture rendering. Specifically, we are developing a research program for a better understanding of the perceptual dimensions associated with texture perception, and the development of algorithms for rendering synthetic textures with desired perceptual qualities.

The first challenge we face is the stability problem of synthetic texture rendering. In particular, there is a need for the specification of the parameter space within which perceptually stable texture rendering can be achieved. Most studies on stability uses a virtual wall as a benchmark of stability performance in haptic rendering. In this case, a haptic interaction can be modeled by a 1 DoF (Degree-of-Freedom) system due to the locally homogeneous geometry of a flat wall. The rendering objective is to make the wall feel as hard as possible without unintended vibrations. Theoretical analysis of stability for interactions with textured surfaces is a much more complex problem. In practice, many authors, such as [8], have commented that the stable range of surface stiffness ensuring *perceptually* stable texture rendering can be quite small. The goal of texture rendering is to evoke sensations related to various aspects of texture perception such as roughness and stickiness [3]. The PHANToM uses the paradigm of feeling through a probe, which means that surface textures are transmitted via temporal cues (as compared to spatial and intensive cues while the bare finger pad is used). From a theoretical points of view, it is difficult to predict what kinds of vibrations would be perceived to be related to texture attributes and what would be perceived as unrealistic. Therefore, we propose to study the parameter space for perceptually stable haptic texture rendering by conducting psychophysical experiments. In this paper, we report results of our study on the range of stiffness parameter that ensures perceptually stable texture rendering, using the method of limits as the experimental paradigm. In Section 2, we describe the experimental design in terms of the rendering methods and the exploration modes employed. The results of the experiments conducted so far are summarized in Section 3, followed by a discussion in Section 4.

*This work was supported in part by a National Science Foundation Faculty Early Career Development Award under Grant No. 9984991-IIS.

2 General Methods

2.1 Apparatus

The hardware setup consists of a PHANToM (Model 1.0A, with encoder gimbals) and a Pentium II PC (400MHz, 128MB RAM). This model of PHANToM has a maximum stiffness of 3.5 N/mm and a workspace of 13 cm × 18 cm × 25 cm.

2.2 Stimuli

The textured surfaces chosen for the stimuli of the experiments are one-dimensional sinusoidal gratings superimposed on an underlying surface. A flat wall is used as the underlying surface in the current study. It is always positioned such that it coincides with the xy plane located at $z = 0$ in the PHANToM coordinate frame. We regard sinusoidal gratings as the basic building blocks for textured surfaces, since any surface profile can be modeled by the weighted sum of sinusoidal functions (see, for example, [7]). The sinusoidal gratings used in this study are described by $z = A\sin(\frac{2\pi}{L}x) + A$, where A and L are the amplitude and the (spatial) wavelength, respectively (see Figure 1).

Two kinds of texture rendering methods are used in the current experiments. Both use a spring model to calculate the magnitude of the rendered force. The first method, introduced by Massie, always generates a force, denoted by $\mathbf{F}_1(t)$, that is normal to the underlying surface [4]. This

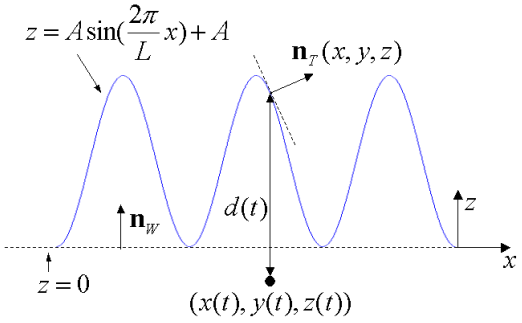


Figure 1: An illustration of the textured surface and its associated variables. See the text in Section 2.2 for details.

method generates forces with a constant direction when the underlying surface is a flat wall. The second method, proposed by Ho, Basdogan and Srinivasan, calculates a force, denoted by $\mathbf{F}_2(t)$, that is normal to the textured surface [2]. This method generates forces that change directions depending on the local micro-geometry of the surface texture.

For both methods, $d(t)$, the penetration depth of the stylus into the textured surface at time t, and the force vectors $\mathbf{F}_1(t)$ and $\mathbf{F}_2(t)$, can be calculated as follows.

$$d(t) = \begin{cases} 0 & , \quad z(t) > 0 \\ A\sin\left(\frac{2\pi}{L}x(t)\right) + A - z(t) & , \quad z(t) \le 0 \end{cases},$$

$$\begin{aligned} \mathbf{F}_1(t) &= K\, d(t)\, \mathbf{n}_W, \\ \mathbf{F}_2(t) &= K\, d(t)\, \mathbf{n}_T(x(t), y(t), z(t)). \end{aligned}$$

where K is the stiffness of the surface, $(x(t), y(t), z(t))$ are the coordinates of the stylus, $\mathbf{n}_W$ is the normal vector of the underlying flat wall, and $\mathbf{n}_T(x, y, z)$ is the normal vector of the textured surface at (x, y, z) (see, again, Figure 1).

2.3 Psychophysical Method

As explained in Section 2.2, the relevant physical parameters are amplitude(A) and spatial period(L) of a sinusoidal grating, and stiffness(K) of the surface.

The goal of our experiments is to quantify the range of the stiffness parameter K where haptically rendered textures feel stable. The method of limits is used for all the experiments reported here. In a typical run, the values of A and L are kept constant, and the value of K is systematically changed (in either ascending or descending order) from trial to trial. The subject's task is to report whether the textured surface feels stable. Each run is terminated when the subject reverses the response from stable to unstable or vice versa. For each experimental condition (i.e., the same pair of A and L values), there are a total of 100 runs with 50 ascending and 50 descending runs. The order of ascending and descending runs is randomized. An ascending run starts with a trial with K_{min}, whose value is chosen to be small enough to ensure a perceptually stable rendering of a textured surface. The subject is instructed to press '1' if the textured surface feels stable, and '2' if it feels unstable. If the subject presses '1', K is increased by a pre-determined value, ΔK, on the next trial. This continues until the subject presses '2'. The value of $K + \Delta K/2$ is then recorded as the estimated threshold for this run. Descending runs are conducted in a similar fashion.

For each experimental condition, the mean and standard deviation of the estimated thresholds of K for the 50 ascending runs (K_A), the 50 descending runs (K_D), and all 100 runs (K_T) are computed and stored separately. Let the lower and upper bounds be $K_L = \min\{K_A, K_D\}$ and $K_U = \max\{K_A, K_D\}$, respectively. The ranges $[0, K_L]$, $[K_U, \infty]$ and $[K_L, K_U]$ correspond to perceptually stable, perceptually unstable, and perceptually 'gray' regions, respectively.

2.4 Experimental Conditions

Four experiments were conducted using the two texture rendering methods and the two exploration modes described below (Table 1):

- **Texture Rendering Methods:** The two texture rendering force computation methods proposed by Massie and Ho et al. (Section 2.2) are used in the experiments. It was observed during preliminary experiments that the two methods produced perceptually distinctive textured surfaces, given the same parameter values. It is therefore necessary to compare their stability characteristics quantitatively.
- **Exploration Modes:** It was also observed during the preliminary experiments that perceived stability of textured surfaces depended on the manner with which the stylus interacted with the surfaces. During the main experiments, the subject is allowed to use two exploration methods: free exploration, or stroking. In the free exploration mode, the subjects can explore the surface in whatever patterns they think are appropriate to discover unrealistic vibrations. In the stroking mode, the subjects are only allowed to move the stylus laterally across the textured surfaces. The stroking mode is particularly interesting because it is the most natural (and the most frequently used) way for gathering texture information through a probe, and because it seems to result in a more stable rendering of textured surfaces.

For each of the four experiments, three values of A (0.5, 1.0, 2.0 mm) and three values of L (1.0, 2.0, 4.0 mm) are used. This results in nine conditions per experiment. The order of the experimental conditions are randomized. For each combination of A and L values, K can vary from 0.0 N/mm (K_{min}) to 0.6 N/mm (K_{max}). This value of K_{max} is chosen to be the maximum stiffness for simulating a stable (not textured) virtual wall, as recommended in the GHOST Programmer's Guide [9]. For ascending or descending runs, K starts from 0.0 N/mm or 0.6 N/mm, respectively. The same increment of ΔK (0.02 N/mm) is used for both ascending and descending runs. Throughout the experiments, the subject is instructed to look away from the PHANToM, and wore headphones through which white noise was played.

Experiment Number	Texture Rendering Method	Exploration Mode
1	$\mathbf{F}_1(t)$	Free Exploration
2	$\mathbf{F}_1(t)$	Stroking
3	$\mathbf{F}_2(t)$	Free Exploration
4	$\mathbf{F}_2(t)$	Stroking

Table 1: Experimental conditions

2.5 Subject

So far, data have been collected for one subject, the first author. This subject has no sensory abnormalities and has considerable experience with haptic interfaces, especially the PHANToM.

3 Results

Typical data for one experimental condition is shown in Figure 2. As is expected with the method of limits, ascending runs tend to generate overestimated thresholds (top panel), and descending runs underestimated ones (middle panel). A combined histogram for ascending and descending runs usually shows a bimodal distribution. This tendency of $K_A > K_D$ has been observed for most of our data, except when both K_A and K_D are very small. The means of the data in Figure 2 are 0.26, 0.19, and 0.23 N/mm for the ascending, descending, and combined runs, respectively. The corresponding standard deviations are 0.04, 0.03, and 0.05 N/mm, respectively. For the remainder of this paper, only the mean and standard deviation for the combined runs are reported.

Data for Experiments 1 and 2 are shown in Figure 3. The same data are plotted as a function of L (top panel) and as a function of A (bottom panel). The two plot symbols, squares and crosses, correspond to data from Experiments 1 and 2, respectively. Equivalently, the same two symbols correspond to data collected using the free exploration and stroking modes, respectively. Several observations can be made from Figure 3. First, the range of K values for stable rendering is consistently larger for the stroking mode than for the free exploration mode. This can be seen from the fact that the crosses are above the squares in both panels of Figure 3. Second, the mean K for stable rendering depended on the values of A, but not L, for the range of A and L tested. Third, the values of K_T for the free exploration mode (i.e., the squares) are less than 0.15 N/mm. This results in a much smaller parameter range for the stable rendering of textured surfaces as compared to that for the rendering of (not textured) flat walls. Data for Experiments 3 and 4 are shown in similar fashion in Figure 4. The main difference between Figures 3 and 4 are that the former used $\mathbf{F}_1(t)$ for rendering, and the latter $\mathbf{F}_2(t)$. Again, the stroking mode results in a larger K range for stable rendering than the free exploration mode. However, there is no clear evidence of strong dependence of data points on either L or A.

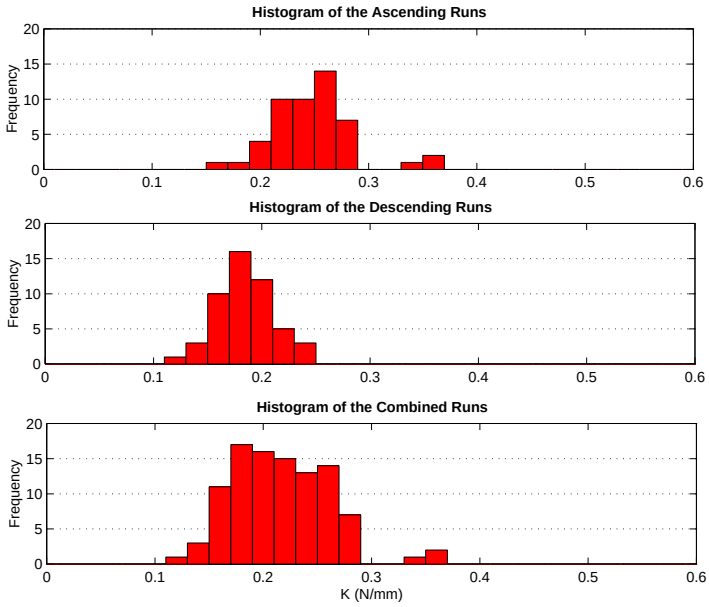


Figure 2: Histogram of the estimated thresholds of K (Experiment 2, $A = 2.0$ mm, and $L = 2.0$ mm)

4 Discussion

In this paper, the method of limits was used to measure the range of K for stable rendering of textured surfaces, using two rendering methods and two exploration modes. Overall, the data demonstrate that the stroking mode results in a more stable rendering of textured surfaces than the free exploration mode, although the effect is much bigger when $\mathbf{F}_1(t)$ was used than when $\mathbf{F}_2(t)$ was used. Basically, the range of K that can be used for stable rendering is very small for the free exploration mode, whether $\mathbf{F}_1(t)$ or $\mathbf{F}_2(t)$ was used. However, for the stroking mode, the use of $\mathbf{F}_1(t)$ results in a larger stable range of K than the use of $\mathbf{F}_2(t)$.

It should be pointed out that the decision to call a particular rendering stable or unstable can be highly subjective and depends greatly on the subject's expectation of how the PHANToM should feel like. It also seems to be the case that different strategies can be employed to determine a perceptual criterion for different combinations of rendering methods and exploration modes. For example, in Experiment 1 where $\mathbf{F}_1(t)$ and free exploration mode were used, the subject judged instability by detecting a vibration when the stylus barely contacted the peaks of the sinusoidal gratings. In Experiment 3 where $\mathbf{F}_2(t)$ and free exploration mode were used, the subject judged instability by paying attention to vibrations when the stylus was in contact with the peaks, or deep inside the valleys of the sinusoidal gratings. It was believed that when the stylus was deep inside the valley, instability could result from the continually changing force directions. For Experiment 2 and 4 where stroking mode was employed, the subject judged instability by whether the perceived bumpiness was mixed with additional vibrations.

Future work will assess the range of stable rendering for

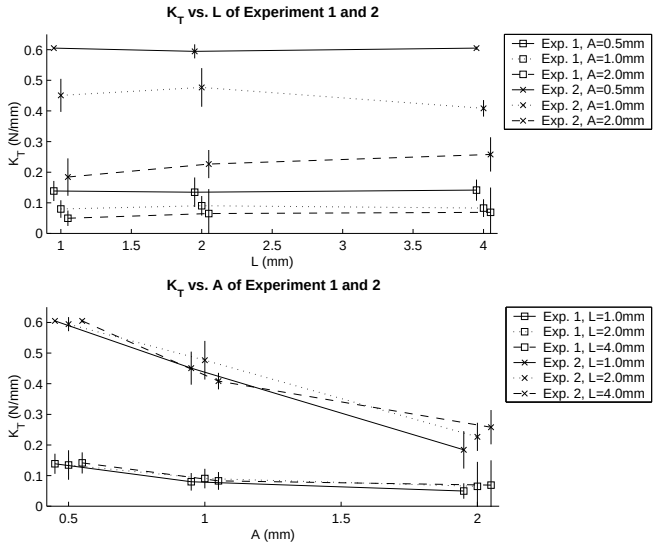


Figure 3: Results of Experiment 1 and 2.
Datum points are slightly offset for a particular L or A value for clarity. Vertical bars show ± 1 standard deviation.

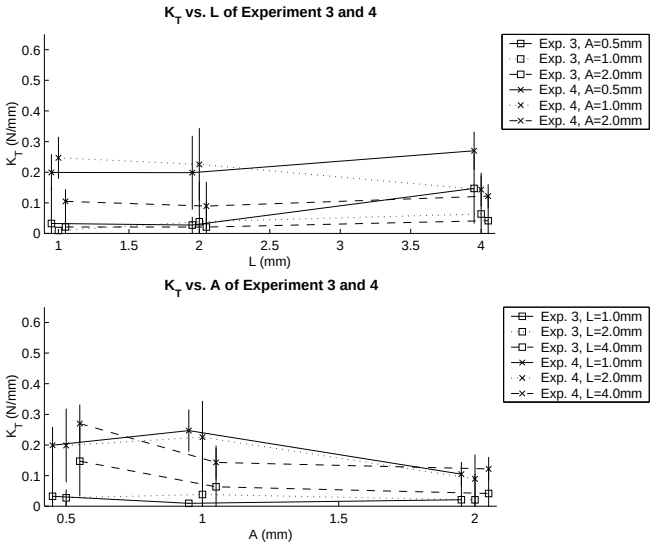


Figure 4: Results of Experiment 3 and 4.
Datum points are slightly offset for a particular L or A value for clarity. Vertical bars show ± 1 standard deviation.

the parameters A and L with additional subjects. Our goal is to define the volume within the three-dimensional space (A, L, and K) where stable rendering of textured surfaces can be achieved.

References

[1] J. P. Fritz and K. E. Barner, Stochastic models for haptic texture, *Proceedings of SPIE's international symposium on intelligent systems and advanced manufacturing - Telemanipulator and telepresence technologies III*, Boston, MA, November 1996

[2] C. Ho, C. Basdogan, M. A. Srinivasan, Efficient point-based rendering techniques for haptic display of virtual objects, *Presence*, Vol. 8, No. 5, pages 477-491, 1999

[3] M. Hollins, R. Faldowski, R. Rao, and F. Young, Perceptual dimensions of tactile surfaced texture: A multidimensional scaling analysis, *Perception & Psychophysics*, 54, 697-705, 1993

[4] T. H. Massie, *Initial haptic explorations with the Phantom: Virtual touch through point interaction*, Master Thesis, MIT, 1996

[5] M. D. R. Minsky, *Computation Haptics: the Sandpaper system for synthesizing texture for a force-feedback display*, Ph D. Thesis, MIT, 1995

[6] J. Siira and D. K. Pai, Haptic texturing - a stochastic approach, *Proceedings of IEEE international conference on robotics and automation*, pp. 557-562, 1996

[7] S. A. Wall and W. S. Harwin, Modeling of surface identifying characteristics using Fourier series, *Proceedings of the ASME dynamic systems and control division*, pp. 65-71, 1999

[8] J. M. Weisenberger, M. J. Krier, and M. A. Rinker, Judging the orientation of sinusoidal and square-wave virtual gratings presented via 2-DOF and 3-DOF haptic interfaces, *Haptics-e*, http://www.haptics-e.org, Vol. 1, No. 4, 2000

[9] Sensable Technologies, Inc., *GHOST SDK Programmers Guide Ver 3.0*, 1999

Pressure Masks for Point-like Contact with Elastic Models

Doug L. James[1] and Dinesh K. Pai[1,2]

University of British Columbia

Abstract

In this paper, we introduce pressure masks for supporting the convenient abstraction of localized scale-specific point-like contact with a discrete elastic object. While these masks may be defined for any elastic model, special attention is given to the case of point-like contact with precomputed linear elastostatic models for purposes of haptic force-feedback.

1 Introduction

It has long been recognized that point contact is a convenient abstraction for haptic interactions, and the PHANToM™ haptic interface is a testament to that fact. While it is possible to consider the contact area to be truly a point for rigid models, this is not possible for elastic models, as infinite contact pressure can lead to various inconsistencies. The solution is simply to assume the contact zone has tractions distributed over a finite surface area. We propose to do this efficiently and consistently by introducing *pressure masks* for defining nodal traction distributions. This addresses at least two core issues. First, having a point contact with force distributed over a finite area is somewhat contradictory, and the traction distribution is effectively an underdetermined quantity without any inherent spatial scale. This is resolved by treating the contact as a single displacement constraint whose traction distribution enters as a user (or manipulandum) specified parameter. The distribution of force on the surface of the model can then be consistently specified in a fashion which is independent of the scale of the mesh. Second, given the model is discrete, special care must be taken to ensure a sufficiently regular force response on the surface, since irregularities are very noticeable during sliding contact motions. By suitably interpolating nodal force responses, displacement constraints can be imposed which will result in regular haptic force-feedback.

The pressure mask approach is particularly effective for haptics when used with linear elastostatic models with precomputed Green's functions, since force response can usually be computed at $\mathcal{O}(1)$ cost. In §2, minimal definitions and notation for discussing the elastostatic model are presented. Afterwards, in §3, the construction and definition of pressure masks is given, and it is shown how to compute nodal (or vertex) stiffnesses for elastostatic models and then use these to consistently define the surface's stiffness.

Much of this material is presented in much greater detail in [JP]; throughout, an identical notation is used.

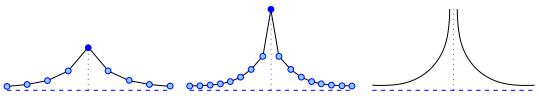


Figure 1: *Point Contact Must Not be Taken Literally for Elastic Models :* This figure illustrates the development of a displacement singularity associated with a concentrated surface force as the continuum limit is approached. In the left image, a unit force applied to a vertex of a discrete elastic model results in a finite vertex displacement. As the model's mesh is refined (middle and right image), the same concentrated force load eventually tends to produce a singular displacement at the contact location, and the stiffness of any single vertex approaches zero (see Table 6).

2 Linear Elastostatic Model Background

Precomputed linear elastostatic models of various discretization origins are efficient candidates for real time haptic interaction [BC96, JP99, JP]. A general boundary Green's function description is now very briefly presented for use in §3.

2.1 Nodal Displacement and Traction Variables

Consider a discrete elastic model with n surface nodes, e.g., polyhedral mesh vertices, for which nodal quantities are defined. Specifically, let the surface displacement $\mathbf{u}(\mathbf{x})$ and traction $\mathbf{p}(\mathbf{x})$ fields be parametrized by n-vectors of nodal variables,

$$\mathsf{u} = [\mathsf{u}_1, \ldots, \mathsf{u}_n]^T \quad (1)$$

$$\mathsf{p} = [\mathsf{p}_1, \ldots, \mathsf{p}_n]^T, \quad (2)$$

where each of the values u_k and p_k belong to $\mathbb{R}^3$. Since our boundary element implementation uses vertex-based triangle mesh models, we shall often refer to a node as a vertex.

[1]Institute of Applied Mathematics

[2]Dept. of Computer Science, {djames|pai}@cs.ubc.ca

2.2 Reference Boundary Value Problem (RBVP) Definition

A major benefit of using linear elastostatic models for haptics is that it is possible to precompute the Green's functions to one particular class of boundary value problem (BVP), a relevant *reference BVP* (RBVP), and be able to efficiently compute components of those solutions rapidly at run time (see Figure 2).

Without loss of generality, assume that either position or traction constraints are specified at each boundary node. Let the mutually exclusive nodal index sets Λ_u^0 and Λ_p^0 specify nodes with displacement and traction constraints, respectively, so that $\Lambda_u^0 \cap \Lambda_p^0 = \emptyset$ and $\Lambda_u^0 \cup \Lambda_p^0 = \{1, 2, ..., n\}$. Specifying boundary values at each of the n nodes defines a BVP to be solved for desired unknown variables, e.g., haptic contact forces, at each step of the simulation. Denote the unspecified and complementary specified nodal variables by

$$\mathsf{v}_j = \begin{cases} \mathsf{p}_j & : \quad j \in \Lambda_u^0 \\ \mathsf{u}_j & : \quad j \in \Lambda_p^0 \end{cases} \quad \text{and} \quad \bar{\mathsf{v}}_j = \begin{cases} \bar{\mathsf{u}}_j & : \quad j \in \Lambda_u^0 \\ \bar{\mathsf{p}}_j & : \quad j \in \Lambda_p^0 \end{cases} \tag{3}$$

respectively.

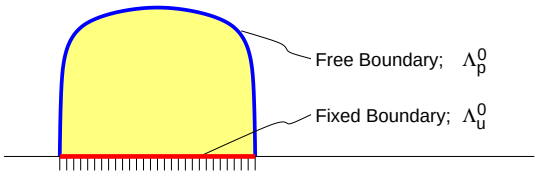


Figure 2: *Reference Boundary Value Problem (RBVP) Example:* The RBVP associated with a model attached to a rigid support is shown with boundary regions having fixed (Λ_u^0) or free (Λ_p^0) nodal constraints indicated. A typical haptic simulation would then impose contacts on the free boundary nodes, Λ_p^0.

2.3 RBVP Solution using Green's Functions

The general solution of the RBVP is conveniently expressed using Green's functions of the RBVP as

$$\boxed{\mathsf{v} = \Xi \bar{\mathsf{v}} = \sum_{j \in \Lambda_u^0} \xi_j \bar{\mathsf{u}}_j + \sum_{j \in \Lambda_p^0} \xi_j \bar{\mathsf{p}}_j,} \tag{4}$$

where the *reference system Green's functions* (RSGFs) are the block columns of the matrix

$$\boxed{\Xi = [\xi_1 \xi_2 \cdots \xi_n] \in \mathbb{R}^{3n \times 3n}.} \tag{5}$$

The j^{th} RSGF describes the effect of the j^{th} node's specified boundary value, $\bar{\mathsf{v}}_j$. In practice it is only necessary to compute RSGFs for nodes which may have changing nonzero boundary values during the simulation.

Since the RSGFs only depend on the RBVP and the geometric and material properties of the deformable object, they may be precomputed for use in a simulation. Note that this applies to any discrete linear elastostatic model, regardless of internal material properties or the discretization technique employed.

3 Surface Stiffness Models for Point-like Contact

This section presents the pressure mask approach for elastic models (§3.1), then specializes to linear elastostatic models for which the pressure masks may be used to compute vertex stiffnesses (§3.2) which are in turn used to compute the surface stiffness (§3.3).

3.1 Pressure Masks for Distributed Point-like Contacts

In this section, pressure masks are defined and used to specify the traction distribution associated with force applied via a masked vertex constraint.

3.1.1 Discrete Traction Space Definitions

In order to characterize traction distributions for the discussion of mask construction and the smoothness of force response, it is necessary to define a discrete scalar function space, $\mathcal{L}$, on the model's boundary, Γ. Let

$$\mathcal{L} = \text{span}\left\{\phi_j(\mathbf{x}), \quad j = 1 \ldots n, \quad \mathbf{x} \in \Gamma\right\}, \tag{6}$$

where $\phi_j(\mathbf{x})$ is a scalar basis function associated with the j^{th} node. The traction field is then a vector function whose components lie in $\mathcal{L}$,

$$\mathbf{p} = \mathbf{p}(\mathbf{x}) = \sum_{j=1}^{n} \phi_j(\mathbf{x}) \mathsf{p}_j.$$

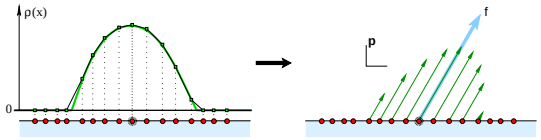


Figure 3: *Collocated Scalar Masks:* A direct means for obtaining a relative pressure amplitude distribution about each node, is to employ a user-specified scalar functional of the desired spatial scale. The scalar pressure mask is then given by nodal collocation (left), after which the vector traction distribution associated with a nodal point load is then computed as the product of the applied force vector and the (compactly supported) scalar mask (right).

3.1.2 Pressure Mask Definition

Scalar relative pressure masks provide a flexible means for modeling vector pressure distributions associated with each node. This allows a force applied at the i^{th}

node to generate a traction distribution which is a linear combination of $\{\phi_j(\mathbf{x})\}$ and not just $\phi_i(\mathbf{x})$.

In the continuous setting, a scalar surface density $\rho(\mathbf{x}) : \Gamma \to \mathbb{R}$ will relate the localized contact force $\mathbf{f}$ to the applied traction $\mathbf{p}$ via

$$\mathbf{p}(\mathbf{x}) = \rho(\mathbf{x})\mathbf{f}$$

which in turn implies the normalization condition

$$\int_\Gamma \rho(\mathbf{x}) d\Gamma_{\mathbf{x}} = 1. \tag{7}$$

In the discrete setting, the surface density on Γ is

$$\rho(\mathbf{x}) = \sum_{j=1}^{n} \phi_j(\mathbf{x})\rho_j \in \mathcal{L}, \tag{8}$$

and is parameterized by scalar pressure mask vector,

$$\rho = [\rho_1, \rho_2, \ldots, \rho_n]^T.$$

Substituting (8) into (7), the discrete normalization condition satisfied becomes

$$a^T \rho = 1, \tag{9}$$

where

$$a_i = \int_\Gamma \phi_i(\mathbf{x}) d\Gamma_{\mathbf{x}} \tag{10}$$

defines the *vertex area*. Notice that the mask density ρ has units of $\frac{1}{\text{area}}$.

In practice, the vertex pressure mask ρ may be specified in a variety of ways. It could be specified at runtime, e.g., as the byproduct of a physical contact mechanics solution, or be a user specified quantity. We shall consider the case where there is a compactly supported scalar function $\rho(\mathbf{x})$ specified at each vertex on the free boundary. The corresponding pressure mask ρ may then be defined using nodal collocation (see Figure 3),

$$\rho_j = \begin{cases} \rho(\mathbf{x}_j), & j \in \Lambda_p^0, \\ 0, & j \in \Lambda_u^0. \end{cases},$$

followed by suitable normalization to satisfy (9).

In the following, denote the density mask for the i^{th} vertex by the n-vector ρ^i, with nonzero values being indicated by the set of masked nodal indices $\mathcal{M}_i$. Since the intention is to distribute force on the free boundary, masks will only be defined for $i \in \Lambda_p^0$. Additionally, these masks will only involve nodes on the free boundary, $\mathcal{M}_i \subset \Lambda_p^0$, as well as be nonempty, $|\mathcal{M}_i| > 0$.

3.1.3 Example: Spherical Mask Functionals

Spherically symmetric mask functionals with a scale parameter were suitable candidates for constructing vertex masks via collocation on smooth surfaces. One example, which was commonly used (see Figure 4 and 6), is a functional with linear radial dependence,

$$\rho^i(\mathbf{x}; r) = \begin{cases} 1 - \frac{|\mathbf{x}-\mathbf{x}_i|}{r}, & |\mathbf{x} - \mathbf{x}_i| < r, \\ 0, & \textit{otherwise}. \end{cases},$$

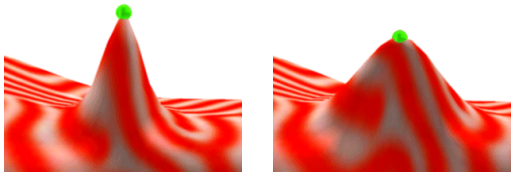


Figure 4: *Illustration of Changing Mask Scale:* Exaggerated pulling deformations clearly illustrate different spatial scales in the underlying traction distribution. In each case, pressure masks were automatically generated using the linear spherical mask functional (see §3.1.3) for different values of the radius parameter, r. This example shows (left) a single vertex mask, and (right) a mask involving several nearby vertices. Note that in each case the surface has been once refined using Loop subdivision.

where r specifies the radial scale[1]. The effect of changing r is shown in Figure 4.

3.2 Vertex Stiffnesses using Pressure Masks

Having consistently characterized point-like force loads using vertex pressure masks, it is now possible to calculate the stiffness of each vertex. In the following sections, these vertex stiffnesses will then be used to compute the stiffness at any point on model's surface for haptic rendering of point-like contact.

3.2.1 Elastic Vertex Stiffness

For any single node, i, on the free, $i \in \Lambda_p^0$, or rigidly fixed boundary, $i \in \Lambda_u^0$, a finite force stiffness, $\mathsf{K}_i \in \mathbb{R}^{3\times 3}$, may be associated with its displacement, i.e.,

$$\mathbf{f} = \mathsf{K}_i \mathsf{u}_i, \quad i \in \Lambda_p^0.$$

Given a force $\mathbf{f}$ applied at vertex $i \in \Lambda_p^0$, the corresponding distributed traction constraints are

$$\mathsf{p}_j = \rho_j^i \mathbf{f}. \tag{11}$$

Then using (4), the displacement of the i^{th} vertex is

$$\mathsf{u}_i = \sum_{j \in \mathcal{M}_i} \Xi_{ij} \mathsf{p}_j = \sum_{j \in \mathcal{M}_i} \rho_j^i \Xi_{ij} \mathbf{f},$$

so that the effective stiffness of the masked vertex is

$$\boxed{\mathsf{K}_i = \left(\sum_{j \in \mathcal{M}_i} \rho_j^i \Xi_{ij} \right)^{-1}, \quad i \in \Lambda_p^0.} \tag{12}$$

It follows from (4) and (11) that the corresponding globally consistent solution is

$$\mathsf{v} = \zeta_i \mathbf{f} = \left(\sum_{j \in \mathcal{M}_i} \rho_j^i \xi_j \right) \mathbf{f} \tag{13}$$

[1] r may be thought of as the size of the haptic probe's tip.

where ζ_i is the convolution of the RSGFs with the mask ρ^i, and characterizes the distributed force load.

# Vertices	Single $\|\mathsf{K}\|_2$	Masked $\|\mathsf{K}\|_2$
34	7.3	13.3
130	2.8	11.8
514	1.1	11.2

Figure 5: *Vertex Stiffness Dependence on Mesh Resolution:* This table shows vertex stiffness magnitudes (arbitrary units) for a BEM model at three different Loop subdivision mesh resolutions. The stiffness corresponding to a single vertex constraint exhibits a large dependence on mesh resolution, and has a magnitude which rapidly decreases to zero as the mesh is refined. On the other hand, the stiffness generated using a vertex pressure mask (collocated linear sphere functional (see §3.1.3) with radius equal to the coarsest mesh's mean edge length) has substantially less mesh dependence, and quickly approaches a nonzero value.

3.2.2 Rigid Vertex Stiffness

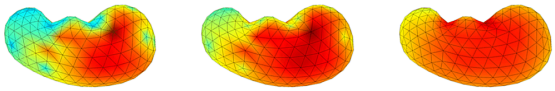
For surfaces of rigid models, a finite force response may be defined using an isotropic stiffness matrix,

$$\mathsf{K}^R = k^{\text{Rigid}}\mathsf{I}_3 \in \mathbb{R}^{3\times 3}, \quad k^{\text{Rigid}} \in \mathbb{R}.$$

This is useful for defining a response at position constrained vertices of a deformable model,

$$\mathsf{K}_j = \mathsf{K}^R, \quad j \in \Lambda_u^0, \tag{14}$$

for determining contact responses on neighbouring triangles which are not rigid.



(a) $a(\mathbf{x})$ (b) $\|\mathsf{K}(\mathbf{x})\|_2$ (c) $\|\mathsf{K}(\mathbf{x})\|_2$

Figure 6: *Effect of Pressure Masks on Surface Stiffness:* Even models with reasonable mesh quality, such as this simple BEM kidney model, can exhibit haptically perceptible surface stiffness irregularities when single-vertex stiffnesses are used. A plot (a) of the vertex area, a, clearly indicates regions of large (dark red) and small (light blue) triangles. In (b) the norm of the single-vertex surface stiffness, $\|\mathsf{K}(\mathbf{x})\|_2$, reveals a noticeable degree of mesh-related stiffness artifacts. On the other hand, the stiffness plotted in (c) was generated using a pressure mask (collocated linear sphere functional (see §3.1.3) of radius twice the mesh's mean edge length) and better approximates the regular force response expected of such a model.

3.3 Surface Stiffness from Vertex Stiffnesses

Given the vertex stiffnesses, $\{\mathsf{K}_j\}_{j=1}^n$, the surface stiffness is defined using nodal interpolation

$$\mathsf{K}(\mathbf{x}) = \sum_{j=1}^{n} \phi_j(\mathbf{x})\mathsf{K}_j, \quad \mathbf{x} \in \Gamma, \tag{15}$$

so that $(\mathsf{K}(\mathbf{x}))_{ij} \in \mathcal{L}$. Note that there are usually only a small number of nonzero terms in the sum of (15). In this way, the surface stiffness may be continuously defined using only $|\Lambda_p^0|$ free boundary vertex stiffnesses and a single rigid stiffness parameter, k^{Rigid}, regardless of the extent of the masks. The benefit of pressure masks is clearly visible in Figure 6 for piecewise linear $\mathcal{L}$.

It follows [JP] that the global deformation corresponding to the displacement constraint $\bar{\mathbf{u}}$ applied on the free boundary at $\mathbf{x} \in \Gamma$ is

$$\mathsf{v} = \sum_{i\in\Lambda_p^0} \zeta_i\phi_i(\mathbf{x})\mathbf{f} = \sum_{i\in\Lambda_p^0} \left(\sum_{j\in\mathcal{M}_i} \rho_j^i \xi_j \right) \phi_i(\mathbf{x})\mathbf{f}. \tag{16}$$

We note that this may be interpreted as an elastostatic generalization of force shading [MS96].

4 Summary and Conclusion

We have introduced pressure masks for the consistent definition of forces arising from point-like haptic interactions. This leads to a computationally efficient means for obtaining regular surface force responses from discrete elastostatic models. Experiments using a PHANToM™ interface confirmed that the pressure masks produced a perceptible improvement.

References

[BC96] Morten Bro-Nielsen and Stephane Cotin. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. *Computer Graphics Forum*, 15(3):57–66, August 1996.

[JP] Doug L. James and Dinesh K. Pai. A unified treatment of elastostatic and rigid contact simulation for real time haptics. To appear.

[JP99] Doug L. James and Dinesh K. Pai. ArtDefo: Accurate Real Time Deformable Objects. *Computer Graphics*, 33(Annual Conference Series):65–72, 1999.

[MS96] Hugh B. Morgenbesser and Mandayam A. Srinivasan. Force shading for haptic shape perception. In *Proceedings of the ASME Dynamics Systems and Control Division*, volume 58, 1996.

Scaling Issues for Teleoperation

Carsten Preusche Gerd Hirzinger
Institute of Robotic and Mechatronic
German Aerospace Center (DLR)
Oberpfaffenhofen, D-82234 Weßling, Germany
E-mail: [Carsten.Preusche,Gerd.Hirzinger]@dlr.de

October 20, 2000

Abstract

In many teleoperation tasks scaling of positions is needed due to different workspaces of the master and the slave robot. Two examples, where the PHANToM T-Model is used for teleoperation with force feedback in our institute, are telesurgery and teleassembly. In both cases the teleoperation task can be divided in an approach phase and a manipulation phase. In the approach phase the PHANToM's movement has to be enlarged, whereas in the manipulation phase the scaling is 1:1 or even zoomed to allow easy manipulation (telesurgery).

In the proposed paper the effects of the position scaling on the forces and/or the control parameters are addressed. As force feedback can give the human operator much aid to fulfill the task, it should not be perturbed due to wrong scaling. Our results show that the human reacts on changes in the stiffness of his environment. So, if scaling is done, the resulting stiffness, that can be detected by the human arm at the PHANToM needs to be the same as without scaling.

1 Introduction

In the field of robotic applications teleoperation plays an important role, due to the still limited autonomous capabilities of robots. Robots equipped with sensors can perform certain tasks in an autonomous way, but their reaction on situations that are not foreseen is limited. Teleoperation is a possibility to enhance the use cases for robots in unknown environments.

Teleoperation has often only visual and acoustic feedback for the operator. Additional sensor information, e.g. forces, can be displayed optionally [HBDH93]. So the operator can not use all his senses to fulfill the task, especially his sensomotoric skills are neglected. Using force feedback will overcome this limitation, so recently a lot of research is done in this field. The PHANToM device [MS94] is one of the first commercial products, that gave a push to these developments.

Introducing a generic haptic device brings up the problem that the master and the slave devices have a different kinematic structure and different workspaces. The problem of different kinematics can be solved by introducing a generic interface, e.g. Cartesian control [OH00]. If the workspace sizes of the master device and the slave robot differ, indexing or scaling has to be done.

To analyze the scaling effects we have two scenarios with different needs and different dimensions. One is the telesurgery, where the PHANToM is coupled with a ZEUS robot arm from ComputerMotion for minimal invasive surgery. Here the interesting space for the surgery lies in a cubic with about 2 cm edge length, but the instrument has to travel through the body about 20 cm. The other case is the teleassembly with an industrial robot from Kuka. The task is to insert a piston into a motorblock, where in the approach phase the piston has to be moved about 1 m and in the put-in phase there is a maximal tolerance of 1/10 mm in the positioning the slave robot. Fig 1 shows the principle setup for the scenarios.

The problem of micro assembly will not be ad-

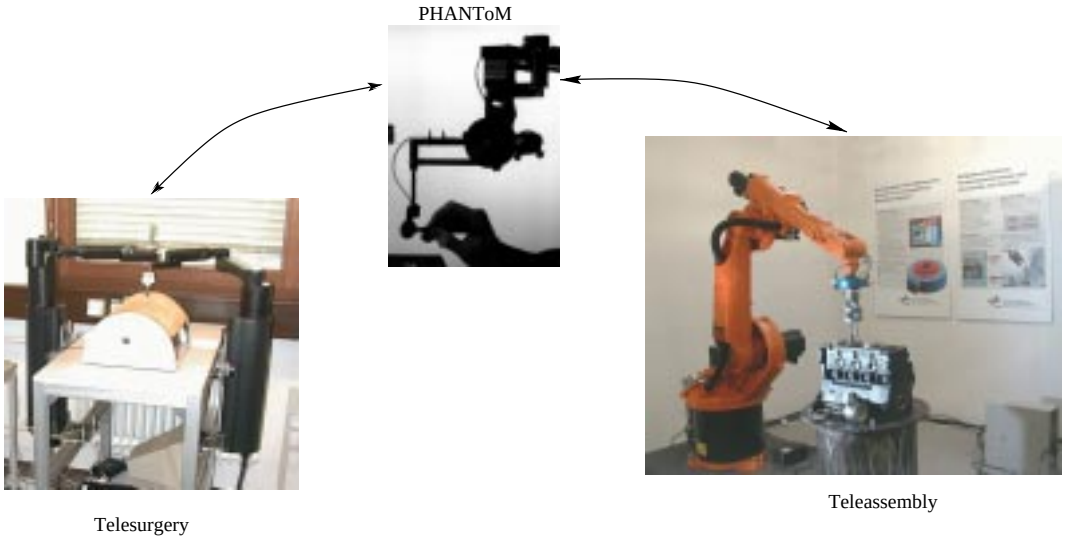


Figure 1: Scenarios for different scaling parameters

dressed. In a micro assembly teleoperation scaling is more difficult, because the sources of the dominant forces, e.g. gravity in the macro world, change [YHUY94].

2 Control Structure

In both scenarios the slave robot is position controlled and the master device has direct force feedback with additional position error feedback. Latter is to stabilize the system due to different dynamic properties of the master and the slave robot. The control structure can be seen in Fig 2.

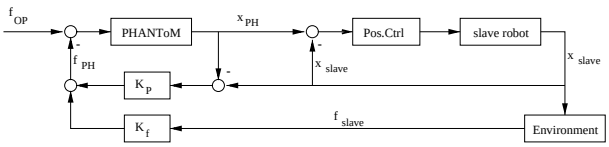


Figure 2: Control structure for teleoperation

The feedback control law is

$$f_{PH} = K_P(x_{PH} - x_{slave}) + K_f f_{Env}, \qquad (1)$$

where x_{PH} and x_{slave} are the positions of the PHANToM and the slave robot respectively, f_{PH} is the force displayed by the PHANToM and f_{Env} is measured at the remote side. K_P and K_f are the control parameters.

K_f is normally equal to one, but can be tuned down, if communication delay affects the stability. K_P represents a virtual coupling between the master and the slave system and depends on the dynamic properties, communication delays and/or the desired task.

3 Indexing

If the workspace of the master device is smaller than the one of the slave robot, only part of the latter workspace can be mapped to the master side, so that it is accessible to the operator. Indexing means, that the movement is not scaled, but that we have a variable offset (index) of the master's position within the slave's workspace, see Fig 3. In this case force feedback is not influenced by the different dimensions of the robots.

The problem of indexing is that it is not very comfortable to move the slave from one manipulation area to another, because e.g. in the teleassembly scenario indexing has to be done very often.

On the other hand, if the interesting area for the manipulation is smaller than the master's workspace (e.g. telesurgery), indexing does not

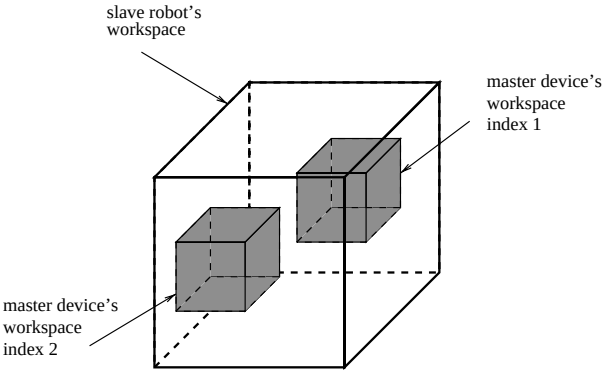


Figure 3: Indexing within slave robot's workspace

help to improve the task.

4 Scaling the movement

A solution to these problems is to scale the movement according to the desired task and task phase, i.e. in an approach phase the master's motion is magnified to the slave's workspace and in the manipulation phase it is kept constant or downsized. Now the effect of this scaling is analyzed. We define the scaling factor s as follows

$$x_{slave} = s \cdot x_{PH}. \tag{2}$$

If scaling is done to the position only, the forces felt by the human keep the same as measured at the remote side and so the appearing stiffness of the environment changes due to the scaling factor s.

$$K_u = \frac{f_{Env}}{\Delta x_{slave} \cdot \frac{1}{s}} = s \cdot K_{Env} \tag{3}$$

where K_u is the stiffness felt at the operator side (neglecting dynamic effects like damping or inertia) , f_{Env} is the force measured and Δx_{slave} is the displacement of the slave robot. K_{Env} is the stiffness of the environment, which is scaled by s to the operator.

If $s > 1$ the stiffness of the environment appears to the human higher as it is and so the system can become unstable. So we have to scale the force with the same factor to avoid this behavior and the resulting feedback law is

$$f_{PH} = K_P(x_{PH} - \frac{1}{s}x_{slave}) + K_f \frac{1}{s} f_{Env}. \tag{4}$$

This case occurs in the teleassembly scenario (Fig 5), because the dimensions of the work cell are much bigger than the workspace of the PHANToM. In the approach phase, when the piston is manoeuvred to the motorblock we use $s = 5$. With this factor still indexing is needed once. But bigger scaling factors resulted in an unstable teleoperation system, because the positional resolution of the human and the dynamic of the robot are limited. During the manipulation phase, i.e. inserting the piston we used the scaling factor $s = 1$ with good results.

If $s < 1$ the motion is scaled down. So the displacement of the slave robot is smaller than the one of the master device. This is equal to the optical zooming. Again as the human feels the properties of the environment like stiffness, these properties should maintain and so the same scaling has to be done to the measured forces.

Figure 4: Telesurgery experiment with German secretary of research Mrs. Bulmahn

In the telesurgery scenario the workspace of the PHANToM is nearly adequate for the approach phase, so scaling $s = 1$ is used. During the manipulation phase, e.g. cutting soft material with a scalpel, we used a scaling factor $s = 0.3$, which led to pleasing results. Even untrained persons are able to handle the teleoperated scalpel secure, as we presented during a visit of the German secretary for research Mrs. Bulmahn (Fig 4).

Approach phase

Manipulation phase

Figure 5: Teleassembly of piston into a motorblock

5 Conclusions

In several experiments with the two different scenarios it was seen that scaling the position and forces is a good way to match the different workspace sizes of master and slave. If it is done carefully the environment stiffness maintained at the master's side and so the human's sensomotoric skills can be used to fulfill the task. In the telesurgery scenario a down scaling was performed in the manipulation task, whereas in the teleassembly scenario the master's motion was scaled up during the approach phase.

Further work has to be done concerning a smooth zooming. This includes also the problem of scaling when the slave is in contact. Then the proposed force scaling will lead to an force step at the master side, which disturbs the feedback and can cause instability.

6 Acknowledgments

Part of the presented work is funded within the 'SFB 453' Collaborative Research Center 'High-Fidelity Telepresence and Teleaction of the DFG (Deutsche Forschungsgemeinschaft).

References

[Bur96] Grigore C. Burdea. *Force and Touch Feedback for Virtual Reality.* John Wiley & Sons, Inc., 1996.

[Föl94] O. Föllinger. *Regelungstechnik.* Hüthig, 1994.

[HBDH93] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl. Sensor-Based Space Robotics - ROTEX and its Telerobotic Features. *IEEE Trans. on Robotics and Automation*, 9(5):649–663, October 1993.

[MS94] T.H. Massie and J.K. Salisbury. The Phantom Haptic Interface: A Device for Probing Virtual Objects. In *Proceedings of the ASME International Mechanical Engineering Congress and Exhibition*, pages 295–302, Chicago, 1994.

[OH00] Tobias Ortmaier and Gerd Hirzinger. Cartesian Control in the dlr minimally invasive surgery telepresence scenario. In *Proceedings of IEEE/RJS International Conference on Intelligent Robots and Systems*, 2000.

[YHUY94] Y. Yokokohji, N. Hosotani, J. Ueda, and T. Yoshikawa. A Micro Teleoperation System for Compensating Scaling Effects Based on Environment Model. In *Proc. of 1999 Japan-USA Symp. on Flexible Automation*, 1994.

Using Haptics and Sound in a Virtual Gallery

Bridget Baird
Center for Arts & Technology
Connecticut College

Introduction

Galleries are traditionally places for visual exploration of objects; concert halls provide auditory exploration. The tools of virtual reality allow for a new kind of gallery: one that encompasses features of a traditional visual museum, means for auditory discovery, and in addition, haptic exploration. The user is invited to browse through this virtual gallery, interacting with the objects, feeling their textures, listening to their audio properties, moving around and inside them. All of this takes place in an interactive, 3D environment where the user navigates and explores with her eyes, ears, and hands.

The Center for Arts and Technology at Connecticut College has provided an atmosphere for interactive collaboration. In the past, many of our projects have involved the interaction between a 3D, visually rich world, and sound. The possibilities for haptic exploration in such a setting were very intriguing, not just for the sake of expanding our digital sensory capabilities to include touch, but also because of the opportunity to explore the interactions and reinforcement among the senses.

Description

The virtual gallery is set inside a hemisphere, with multi-modal sculptures placed near the edge. From the center, where the user enters, the objects are not all clearly visible, although light and vague forms give clues to their presence. At the center of the world the user experiences a dark and foggy atmosphere, with rumbling sounds. As she moves towards the edges of the world there is more light as well as visual and auditory clarity. Nearing an object causes it to come into focus, both visually and through sound. The rumbling noise that she experiences in the center of the world lessens and begins to blend with the individual auditory properties associated with the object itself. Navigation is by means of a joystick, which allows her to move forward or backward, up or down, and yaw from side to side. The user also has stereo eyeglasses, which increase the feeling of immersion in the three-dimensional world. The user's right hand controls a phantom. The model for navigation is that of a person walking through a world, holding a hand out in front. In this virtual world there is no gravity, and both the objects and the user can float in space. Other laws of physics are partially observed, depending on the object: some can be bumped into with the phantom, some are impenetrable, others can be entered.

The virtual gallery is programmed in Visual C++ on a PC and uses Sense8's World Toolkit libraries as well as the Ghost 3 libraries. Models were created in a dxf format and then translated to VRML2, for both the haptic and visual worlds. Because of the addition of the World Toolkit libraries it was necessary to create dual haptic and graphic worlds.

Haptic Interactions

Each of the five objects possesses visual, audio, and tactile properties. As the user nears the object, the sound(s) associated with that object become louder. The audio is spatialized so that

as the user moves around the world, the audio changes. Each object also has haptic characteristics. All of these characteristics interact with each other.

One of the objects is multi-faceted and has many angles on its surfaces. The sound associated with that object is voices which speak and overlap. The surface of the object has some give to it so the user can not only feel the object, but push on it. It is also possible to push through and enter the object, either exploring solely with the phantom, or moving hand and head into the object. If the user enters this object with her head (ears), the audio files exhibit reverberation and the voices seem to echo and bounce off the interior walls. On the inside of the object the user is also able to explore or push back through to the outside. Another of the objects has two moving arms mounted on a central cylinder. Contact with the arms causes them to start swinging and also triggers their individualized sounds. The arms, because they can be in motion, have no haptic properties, although the central cylinder does. Another object consists of interlocking rings. When the user enters the negative space of the holes in the rings she experiences a haptic buzzing, which is accompanied by a sound. The rings themselves have a rather hard surface and as the user presses on them with the phantom she causes a change to a higher pitch in their sound. Pressure on the rings and the rise in pitch are correlated on a sliding scale. One of the other objects is in motion as the user approaches, with the haptics turned off. When the user touches it (as determined in the graphics world) the haptics are activated and the user is able to explore its surface. Another object has hollow bumps or hemispheres on its surface and as the user goes up inside these hemispheres the viscosity of the phantom increases while at the same time the pitch of the associated sound is lowered. This causes the sensation of moving inside a thick substance where time has slowed. Throughout all of our design we have looked for ways in which the multiple senses can interact and reinforce one another.

Phantom Challenges

The necessity for including virtual reality libraries made it impossible to use the OpenGL Ghost environment. Thus it was necessary to create dual haptic and graphic environments. This situation was made more complex because Ghost and Sense8 have different coordinate conventions.

A more major difficulty occurred in setting up navigation for the phantom. Because the user is free to move around the world, it was necessary to translate the center of the phantom (in the virtual world) and also adjust its direction. Navigation is by means of a joystick held in the left hand; the joystick represents the head (and body) of the user. We restricted navigation to forward/backward, yawing from side to side, and up/down elevator motion (using a button). This navigation seemed to be simple enough to master quite easily, and flexible enough to move the user where she wishes to go with relative ease. The right hand operates the phantom; the visual point of contact (in the virtual world) of the phantom is represented by a small, slightly glowing sphere. The paradigm is that of a human: the right hand can move, but it is attached to the body and so as the body moves through the gallery, the hand moves with it, maintaining its relative position. The difficulty that this presented was that movement with the joystick could easily cause the phantom to come in contact with an object, causing a conflict. Initially this gave rise to a constant vibration in the phantom. The (somewhat OK) programming fix was to first check to see if the phantom is in contact with any of the objects. If it was, we don’t update its

position. If not and if either the magnitude of the translation or the magnitude of directional shift is greater than some fixed lower bounds, update the phantom. Some slight adjustments had to be made when considering the particular object into which the user can enter: in this case we had to set touchableByPhantom to FALSE, move the phantom, and then set it back to TRUE. For all of the objects we still encoucntered the problem of coming up in the middle of an object and thus causing excessive force, but could not find a good way around this. Setting the objects so they can't be felt by the phantom begs the question. We thought of moving the phantom away from the object slightly so that we wouldn't have immediate contact, but then we ran into the problem of what constitutes the inside and the outside of objects, and this question is already vexing enough for our graphics. The problem of navigation is certainly the largest one we encountered and we have not solved it to our satisfaction.

Other problems were more standard. We get low frame rate, even with a dual processor and a graphics card. This is probably because, in addition to the graphics and haptics, we are doing real-time sound processing: spatializing, mixing sounds together, changing pitches, adding reverberations, etc. Another problem is the proliferation of devices: manipulating a world that includes eyeglasses or HMD, sound, joystick and phantom creates a cumbersome environment, but a manageable one.

Future Research

The project we are currently working on is really a test bed for using haptics, vision, and audio in a 3D world. We have been very pleased with the results so far. We are particularly excited about the opportunities that haptics presents in these types of arts and technology collaborations. There are some additional tests we would like to do in the present project to make the interface run more smoothly and to get the frame rate down. Future work of this nature includes a project to explore an abandoned abbey (Tintern) by re-creating the original structure, allowing the user to explore and feel. We will use textures from photographs and we envision the user being able to touch the walls, causing them to crumble into the ruins of today. We have also been talking with colleagues about the possibility of using haptics to "feel" the envelopes of musical sound, manipulating audio properties in an interactive manner. The Center for Arts and Technology is an ideal venue for taking haptics in some of these new directions.

Bridget Baird, Professor of Mathematics and Computer Science
Judith Ammerman '60, Director of the Center for Arts and Technology
Connecticut College
New London, CT 06320
(860) 439-2008
http://math.conncoll.edu/faculty/bbaird
bbbai@conncoll.edu

Collaborators: Ozgur Izmirli programmed the audio portion, David Smalley modeled the 3D objects, and Don Blevins assisted with general programming.

The Visual Haptic Workbench

J. Dean Brederson Milan Ikits Christopher R. Johnson Charles D. Hansen

Scientific Computing and Imaging Institute
School of Computing, University of Utah
50 Central Campus Drive Rm 3490
Salt Lake City, UT 84112-9205, USA
{jdb, ikits, crj, hansen}@cs.utah.edu

Abstract

Multimodal interfaces have been shown to increase user performance for a variety of tasks. We have been investigating the synergistic benefits of haptic scientific visualization using an integrated, semi-immersive virtual environment. The Visual Haptic Workbench provides multimodal interaction; immersion is enhanced by head and hand tracking, haptic feedback, and additional audio cues. We present the motivation and design goals for this system, discuss its current implementation, and describe some initial applications. Preliminary results indicate that visualization combined with haptic rendering intuitively conveys the salient characteristics of scientific data.

1 Introduction

Haptic interfaces have been shown to be advantageous as an interaction modality for specific tasks [25, 2]. They provide bidirectional interaction via position sensing and force feedback, thereby utilizing additional sensory channel bandwidth of the user. By combining haptic rendering and visualization, we hope to increase intuitive understanding of scientific data.

For this purpose, we have designed and implemented a testbed system, comprised of a SensAble PHANToM 3.0 and a Fakespace Immersive Workbench. Using this system, we are investigating the synergistic benefits of combined visual and haptic data rendering.

There are a several important issues to resolve when building such an integrated system. The hardware components must be carefully selected based upon their performance characteristics, ease of integration, and flexibility for future enhancements. The software infrastructure requires a modular design, efficient cooperation between its elements, and performance optimization. Robust calibration and coregistration of the individual device workspaces will enable an integrated environment capable of *bounded error interaction*. This can be quantified as a unified error tolerance which bounds the total system error throughout the physical workspace.

After developing a software infrastructure for our hardware setup, we experimented with synergistic rendering techniques using vector field data. Preliminary results based on informal user evaluation indicate that the Visual Haptic Workbench is an effective tool of discovery for the exploration of scientific datasets.

2 Previous Work

Virtual workbench environments [15] have been employed successfully for a multitude of purposes over the last several years. Similarly, haptics research has produced many interesting results, a recent survey of which can be found in [10]. Much of this research is concerned with geometric interaction, but several applications of haptics to scientific visualization are relevant to the development of our system.

Early integrated systems include Project GROPE [3], the Nanomanipulator [17], and the Nanobench [8]. These projects focused on rendering aspects of molecular dynamics, drug docking simulations, and real-time interactions with a scanned-probe microscope. An Argonne Remote Manipulator and a PHANToM were used as the haptic interfaces, and the visual displays ranged from a desktop monitor to a custom workbench. A related system used a PHANToM with an augmented reality display for examining seismic and geological data [22]. Iwata et al. proposed and implemented simple vector field interactions using a custom HMD and haptic interface [13].

Research on haptic volume rendering used scalar data value and gradient as force model parameters to explore and modify volumetric data [1]. A treatment of haptic rendering and scientific visualization is given in [7], with emphasis on

tools for the visually impaired. Recently, research on combined visual and haptic display methods used a custom haptic interface and commercial visualization software for exploration of certain scientific datasets [12, 16].

While there is a small but growing body of research on this topic, there remains much to discover about the synergistic display of scientific data. The Visual Haptic Workbench is a testbed system for conducting this research.

3 Design Goals and Implementation Issues

Building a multimodal system for synergistic display of scientific data involves identifying a set of performance criteria and dealing with the following implementation issues:

- **Calibration** The visual and haptic workspaces must be accurate to provide faithful data rendering. Ideally, a high-fidelity synergistic display yields a direct mapping between virtual and physical spaces. Position tracking enables user-centered visual display and immersive interaction. Large tracking inaccuracies are unacceptable, as they lead to incorrect stereoscopic projections and inconsistent manipulation and navigation. Haptic interfaces enhance interaction by creating kinesthetic cues via proprioceptive feedback. To avoid perceptual conflicts, they must be kinematically calibrated from reproducible fiducials.

- **Coregistration** Individually calibrated workspaces need to be coregistered to accommodate their relative location, orientation, and scale. By fusing multiple workspaces, a uniform bound can describe position, orientation, and temporal errors in physical space.

- **Compensation** Latency in the system can seriously degrade interactivity, which has been shown to impede user performance [6]. Device communication and computational delays can be compensated by predictive filtering, parallel computation, and a careful runtime mapping of the application to the underlying hardware.

We also considered specific research applications to pursue with our system. At the SCI Institute, a variety of datasets are routinely investigated. Typical examples include volumetric CT/MRI/MRA data, computational fluid dynamics data, and finite element solutions to bioelectric field problems. Recent research has focused on diffusion tensor MRI and multimodal medical data such as fused MRI/EEG volumes [14]. These datasets range in size from megabytes to gigabytes and may be static or time-varying on a variety of grid types. Considering our research needs, the supporting software must be efficient, modular, extensible, and scale well with data size.

4 Current Implementation

We have constructed a prototype system consisting of a SensAble PHANToM 3.0 mounted in a T configuration above a Fakespace Immersive Workbench (see Figure 1). The PHANToM is suspended above the workbench with a cross-braced lumber frame. While not as structurally stiff as desired, it is an inexpensive means to experiment with the design parameters for the final mounting. We replaced the original “push to interrupt” switch with a “step to operate” footswitch as a more convenient safety mechanism. The infrastructure of our system can be described in terms of its hardware components and software architecture.

Figure 1. The Visual Haptic Workbench.

4.1 Hardware Components

The Visual Haptic Workbench consists of five hardware components, as shown in Figure 2. The dominant hand of the user experiences haptic feedback from the PHANToM, and the subdominant hand navigates through a menu interface via Pinch glove contact gestures. Head tracking is done with a Polhemus Fastrak receiver mounted on a pair of Stereographics CrystalEyes LCD shutter glasses. The subdominant hand can also be tracked with a separate receiver to facilitate more complex interaction paradigms. The audio subsystem gives the user additional reinforcement cues to clarify the application interface. Finally, the Immersive Workbench provides a correct stereo perspective view for the user based on the tracked head location. These components are handled by five concurrent processes running on an SGI Onyx2 with 250 MHz R10000 processors and InfiniteReality2 graphics.

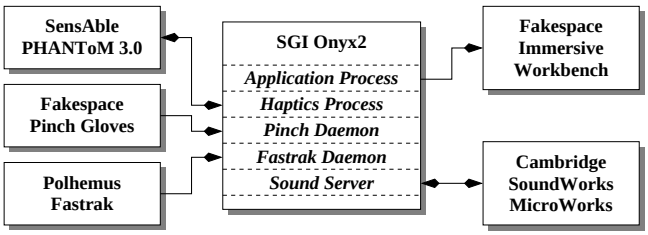


Figure 2. Hardware components of the system.

4.2 Software Architecture

We designed and implemented a software framework using the GHOST SDK for application development on the Visual Haptic Workbench (see Figure 3). The ghostGLManager class is overridden to support semi-immersive virtual environments. This extension is built upon the VGL library, internal VR software that provides graphical rendering and an interface for a variety of tracking and interaction devices. In addition, we derive a set of classes for haptic data rendering techniques from the gstForceField class. These haptic rendering classes are part of a library for synergistic display techniques. The SD Library also contains visualization methods [19, 26], interface widgets, dataset classes, menu functions, and geometry tessellators. The application and haptics processes access common GHOST state through shared memory. We use the NCSA VSS software package for audio support [24]. The application client communicates with the VSS server asynchronously via UDP messages to produce audio feedback. To maintain interactive update rates, each software component runs on a separate processor with appropriate scheduling priority (see Figure 2).

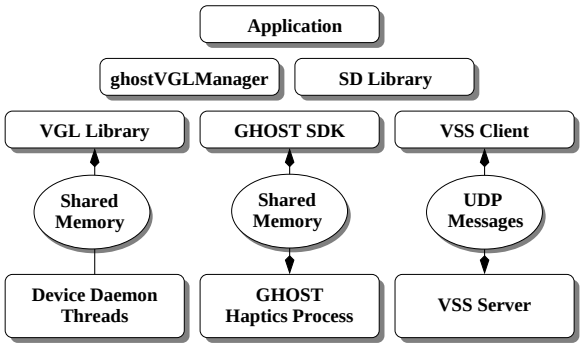


Figure 3. Application software architecture.

5 Initial Evaluation

To evaluate our initial development efforts, we examined two static 3D vector field datasets. One is an analytical electrostatic charge field, and the other is a simulated tornado funnel. Synergistic data rendering is accomplished via advanced visualization techniques [26], and extension of the vector field haptic rendering ideas presented in [21]. Figure 4 shows a user interacting with the simulated tornado dataset.

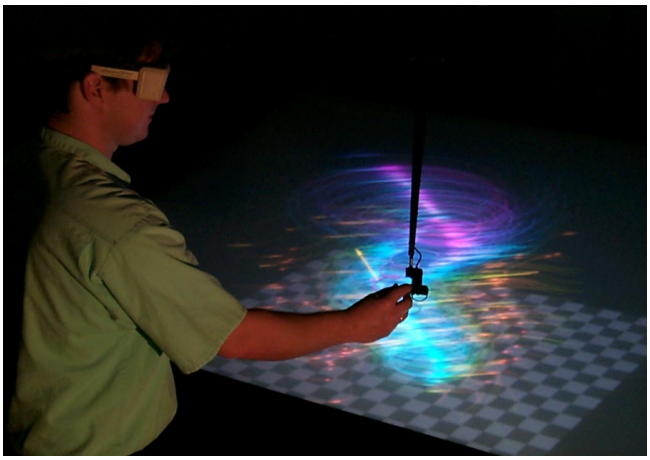


Figure 4. Synergistic data display on the Visual Haptic Workbench.

Our initial implementation addresses some of our design goals, as described in Section 3. Methods for quantifying and correcting magnetic tracker distortion have been developed and are incorporated into our system [11]. The PHANToM uses the standard GHOST "calibration" procedure, which is insufficient to satisfy our performance criteria. The workbench display is driven by an Electrohome 9500LC projector with folded optics and a nonlinear diffusion surface. Unfortunately, these characteristics limit our current display calibration to the adjustment of projector parameters. We coregister the visual and haptic workspaces indirectly by measuring the PHANToM "calibrated" endpoint with a tracked receiver. Detailed analysis of the runtime characteristics of our software, followed by careful matching of hardware devices to machine resources yields a highly interactive application.

To date, we have demonstrated our prototype to over one hundred visitors, the majority of whom found the demo applications compelling. During these demonstrations we observed a variety of new users, whose reactions suggested further improvements to our system.

6 Conclusions and Future Work

The current implementation can be improved in several respects: better tracking technologies, stereo digital projector for improved display calibration, primary surface mirror and linear diffusion material for crisper display, higher per-

formance graphics hardware, 6DOF enhancement for end-effector torques [4], kinematic calibration of the PHANToM, and improved device control for high-fidelity haptics.

Currently, we are experimenting with a variety of volumetric data, including brain MRA and diffusion tensor MRI, and the Visible Human [20]. We are also extending our methods to render a wider variety of flow data, both steady and unsteady, on regular and irregular grids. In addition, we intend to explore methods for quantum molecular dynamics, meteorology, and terrain applications. Our overarching goal is to provide comprehensive synergistic display for a wide variety of scientific data.

In addition to focusing on synergistic display techniques, there are other useful applications to consider. One could interactively segment and register datasets more rapidly than using image-based approaches. Novel applications for computer graphics have already been developed [9, 23, 5]. A characterization of haptic widgets was published in [18], which we intend to extend for our own interface needs. Educational applications could intuitively render haptic manifestations of abstract or unfamiliar concepts. Vector calculus and classical physics are examples of subjects that could be made tangible to students for improved comprehension.

We have designed and built a prototype system for synergistic display of scientific data. By developing and demonstrating initial applications, we have been able to refine our system and identify future research directions. To meet our design goals and address research needs, significant work remains to be done. Nevertheless, the Visual Haptic Workbench is an exciting project and we are compelled to realize its potential.

Acknowledgements

The authors thank Hope M.F. Eksten for her help with various construction and fabrication projects. Support for this research was provided by ARO DURIP grant DAAG-559710065, the DOE Advanced Visualization Technology Center (AVTC), and NSF Grant ACI-9978063.

REFERENCES

[1] R. Avila and L. Sobierajski. A haptic interaction method for volume visualization. In *Proc. IEEE Visualization*, pages 197–204, 1996.

[2] C. Basdogan, C. Ho, M. Slater, and M. Srinivasan. The role of haptic communication in shared virtual environments. In *Proc. Third PHANToM Users Group Workshop*, 1998.

[3] F. Brooks, M. Ouh-Young, J. Batter, and P. Kilpatrick. Project GROPE – Haptic displays for scientific visualization. In *Proc. ACM Siggraph*, pages 177–185, 1990.

[4] E. Chen. Six degree-of-freedom haptic system for desktop virtual prototyping applications. In *Proc. of the First International Workshop on Virtual Reality and Prototyping*, pages 97–106, 1999.

[5] B. Donald and F. Henle. Haptic vector fields for animation motion control. In *Proc. Fourth PHANToM Users Group Workshop*, 1999.

[6] S. Ellis, S. Adelstein, G. Baumeler, G. Jense, and R. Jacoby. Sensor spatial distortion, visual latency, and update rate effects on 3D tracking in virtual environments. In *Proc. IEEE Virtual Reality*, pages 218–221, 1999.

[7] J. Fritz. Haptic rendering techniques for scientific visualization. Master's thesis, University of Delaware, 1996.

[8] B. Grant, A. Helser, and R. Taylor II. Adding force display to a stereoscopic head-tracked projection display. In *Proc. IEEE VRAIS*, pages 81–88, 1998.

[9] A. Gregory, S. Ehmann, and M. Lin. inTouch: Interactive multiresolution modeling and 3D painting with a haptic interface. In *Proc. IEEE Virtual Reality*, pages 45–52, 2000.

[10] J. Hollerbach and D. Johnson. Haptic rendering. In M. Cutkosky, R. Howe, K. Salisbury, and M. Srinivasan, editors, *Human and Machine Haptics*. MIT Press, 2000.

[11] M. Ikits, J. Brederson, C. Hansen, and J. Hollerbach. An improved calibration framework for electromagnetic tracking devices. In *Proc. IEEE Virtual Reality*, 2001 (submitted).

[12] F. Infed, S. Brown, C. Lee, D. Lawrence, A. Dougherty, and L. Pao. Combined visual/haptic rendering modes for scientific visualization. In *Proc. of 8th Annual ASME Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, pages 93–99, 1999.

[13] H. Iwata and H. Noma. Volume haptization. In *Proc. IEEE Symp. Research Frontiers in Virtual Reality*, pages 16–23, 1993.

[14] C. Johnson, S. Parker, C. Hansen, G. Kindlmann, and Y. Livnat. Interactive simulation and visualization. *IEEE Computer*, 32(12):59–65, December 1999.

[15] W. Krüger, C. Bohn, B. Fröhlich, H. Schüth, W. Strauss, and G. Wesche. The responsive workbench: A virtual work environment. *IEEE Computer*, 28(7):42–48, 1995.

[16] D. Lawrence, C. Lee, L. Pao, and R. Novoselov. Shock and vortex visualization using a combined visual/haptic interface. In *Proc. IEEE Visualization*, pages 131–138, 2000.

[17] W. Mark, S. Randolph, M. Finch, J. Van Verth, and R. Taylor II. Adding force feedback to graphics systems: Issues and solutions. In *Proc. ACM Siggraph*, pages 447–452, 1996.

[18] T. Miller and R. Zeleznik. The design of 3D haptic widgets. In *Proc. Symp. on Interactive 3D Graphics*, pages 97–102, 1999.

[19] G. Nielson, H. Hagen, and H. Müller, editors. *Scientific Visualization : Overviews, Methodologies, and Techniques*. IEEE Computer Society Press, 1997.

[20] http://www.nlm.nih.gov/research/visible/visible_human.html.

[21] L. Pao and D. Lawrence. Synergistic visual/haptic computer interfaces. In *Proc. of Japan/USA/Vietnam Workshop on Research and Education in Systems, Computation, and Control Engineering*, pages 155–162, 1998.

[22] D. Stevenson, K. Smith, J. Mclaughlin, C. Gunn, J. Veldkamp, and M. Dixon. Haptic workbench: A multisensory virtual environment. In *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems VI*, volume 3639, pages 356–366, 1999.

[23] http://www.sensable.com/FreeForm/.

[24] http://cage.ncsa.uiuc.edu/adg/VSS/.

[25] S. Wall and W. Harwin. Quantification of the effects of haptic feedback during a motor skills task in a simulated environment. In *Proc. Second PHANToM Users Research Symposium*, 2000.

[26] M. Zöckler, D. Stalling, and H. Hege. Interactive visualization of 3D-vector fields using illuminated streamlines. In *Proc. IEEE Visualization*, pages 179–185, 1996.